TURBO news

REVISTA PARA COMPUTADORES ATARI Nº 7 · ENERO/FEBRERO 1990

$550

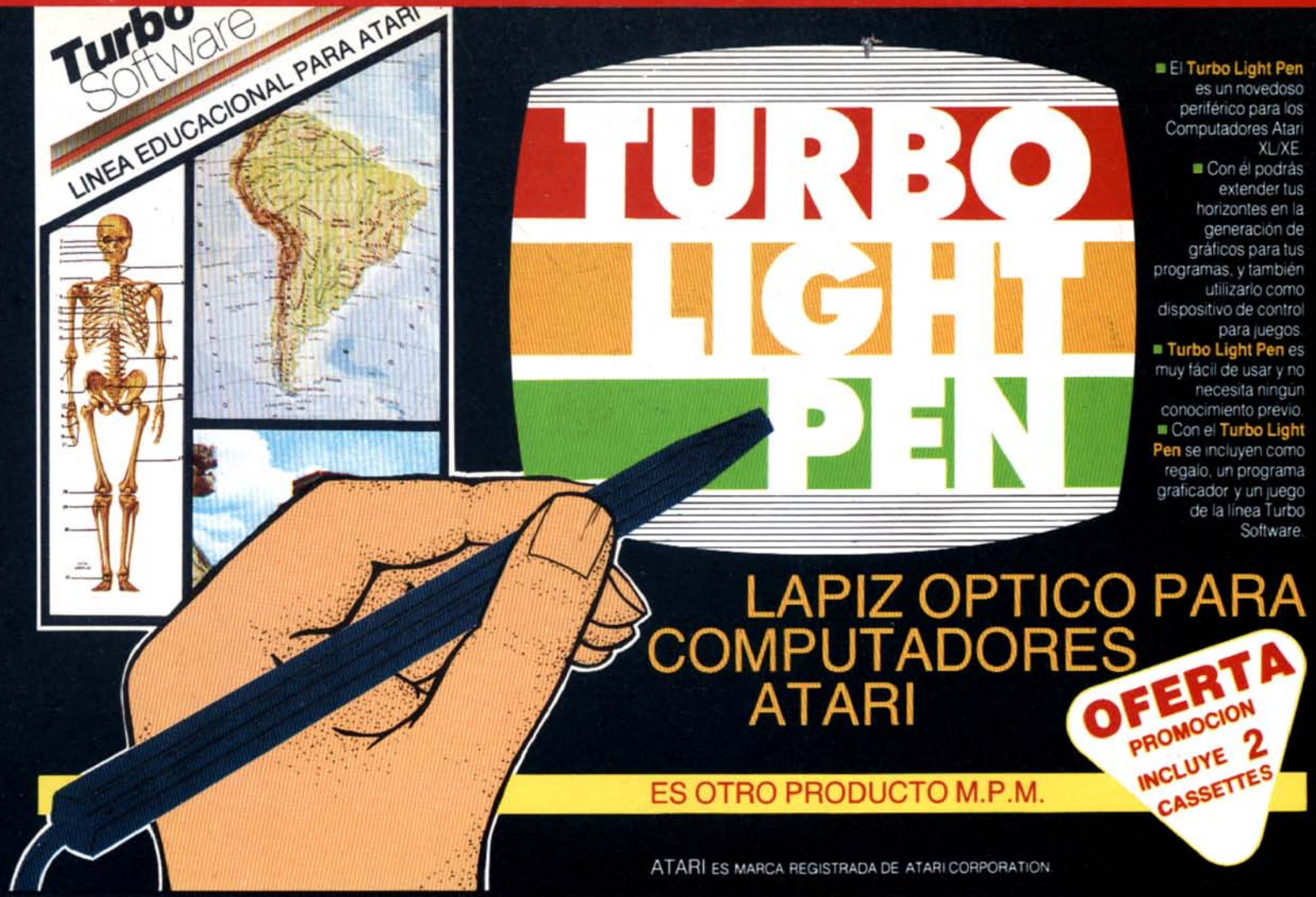
ATARI

OFERTA DE PROMOCION

KIT DE JUEGOS turbo news

INCLUYE:

- 2 JUEGOS
- 1 REVISTA TURBO NEWS



ADQUIERELOS EN LOS SIGUIENTES PUNTOS DE VENTAS

• **ANTOFAGASTA:** COOPERCARAB KW VIDEO LA ESPAÑOLA • **VIÑA DEL MAR:** FALABELLA VIÑA INSIS MPR COMPUTACION • **VALPARAISO:** COMPUTRONIC • **SANTIAGO:** AUDIO BICICLETA INTERNAC CASA ROYAL CENTRO ATARI COMERCIAL ESTADO COMPUMANQUE COMPUCENTER FALABELLA AHUMADA FALABELLA P. ARAUCO IMACO INFOGROUP PC STORE PETERSEN ROLEC SUPERMERCADOS UNIMARC TASCO VIDEO CLUB INTERNACIONAL • **RANCAGUA:** CASA ZUÑIGA • **CURICO:** MULTIHOGAR • **TALCA:** LIBRERIA "EL AHORRO" MULTICENTRO VIDEO CLUB CASSAL • **CHILLAN:** CASA EDISON • **CONCEPCION:** COOPERCARAB DISMAR DISMAR 2 EQUS PHANTER RAPSODIA SESCO • **LOS ANGELES:** DISTRIBUIDORA MERINO • **ANGOL:** SCORPIO • **VICTORIA:** CASA SIGMUND • **TEMUCO:** COMERCIAL MANQUEHUE ESTABLECIMIENTOS GEJMAN FALABELLA • **PUCON:** ELTIT • **VILLARRICA:** JOYERIA KETTERER • **VALDIVIA:** ELECTROMUSICA • **LA UNION:** IMPORTADORA COSMOS • **OSORNO:** CASA REAL FOTO EXPRESS • **PUERTO VARAS:** ELECTRO HORN • **PUERTO MONTT:** COMERCIAL MANQUEHUE

EDITORIAL

Queridos Amigos:

En el número especial de vacaciones de nuestra Revista TURBO news que abarca los meses de enero y febrero, volvemos a la carga con la misma fuerza de siempre, para satisfacer tu curiosidad provocada por este pequeño gran amigo: El Computador Atari.
En este número tenemos grandes novedades, como ser una descripción tanto lógica como física del diskette. En el curso de Assembler iniciamos un gran proyecto que esperamos ir resolviendo junto a ustedes: El desarrollo de un lenguaje nuevo de programación hecho por nosotros en Assembler.

Incorporamos también en este número un Puzzle para que demuestren sus conocimientos de computación. Continuamos por supuesto con los cursos de Basic, Logo y el recientemente incorporado Curso de Gráficos por Computadora. El ranking sigue recibiendo tus preferencias y estamos preparando el Gran Ranking del Año 89.

Estamos seguros que, a pesar de las vacaciones, el computador no dejará de trabajar, así que manos a la obra, continuemos descubriendo sus secretos y nos veremos como siempre mes a mes.

CONTENIDO

2 TURBO MAIL

4 CURSO BASIC (Lección 7)

6 CURSO LOGO (Lección 5)

9 CURSO ASSEMBLER (Lección 7)

13 GRAFICOS POR COMPUTADORA (Segunda parte)

16 EXPLORANDO EL DISKETTE

22 RANKING DEL MES DESCRIPCION DE JUEGOS

25 PROGRAMAS

TURBO news

Circulación mensual, Nacional e Internacional.
Destinada a los usuarios de computadores ATARI (R) como material didáctico de Programación. TURBO news (R) es una publicación de EDITORA TURBO LTDA. Domicilio: Avda. Holanda 2456 – Teléfono: 2238063. SANTIAGO, CHILE.

DIRECTOR RESPONSABLE: Mauro Pieressa. REPRESENTANTE LEGAL: Marcelo Waldbaum. PRODUCCION: Marcelo Waldbaum y Mauro Pieressa, Programadores y Diseñadores de Computación. DIRECTORA DE ARTE: Odalí Guerrero L. PUBLICIDAD Y RR.PP.: Liliana Muñoz Otárola, Hernán Vittini. COLABORACION: Silvia Edelstein. CORRECTOR: Marcial Valenzuela S. FOTOGRAFIA: Catalina Botto. FOTOCOMPOSICION: Brubytes. IMPRESION: Herograf. DISTRIBUCION: Alfa Ltda. Agradecemos la colaboración de COELSA S.A. Centro Atari (Augusto Leguía Sur 75). Profesor Emilio Antileff. Atari es marca registrada de ATARI CORPORATION. TURBO news es marca registrada de EDITORA TURBO LTDA. (Reg. Marca Nº 342428 9-05-89).

Bilbao 4226 486506

TURBO MAIL

TURBO NEWS:

En la edición de noviembre de mi Revista, encontré unas rutinas muy interesantes de sonidos y empezé a investigar nuevas posibilidades.

Es así como les mando dos rutinas de sonidos una de ellas de un helicóptero y la otra de una bandada de pájaros en vuelo.

No puedo dejar pasar esta ocasión, para felicitarlos por el excelente material que trae la Revista TURBO news y sugerirles que incluyan en su revista más programas, sobre todo educacionales y juegos no bélicos.

RUTINA DE HELICOPTERO:

```
10 FOR A=1 TO 20 STEP 2
20 SOUND 0,A,35+A,A
30 NEXT A
40 GOTO 10
```

SONIDO DE BANDADA DE PAJAROS:

```
10 FOR A=1 TO 45
20 FOR B=1 TO 20 STEP 2
30 SOUND 0,11,A+B,A+O.5
40 SOUND 1,A,B,A+11
50 NEXT B
60 NEXT A
70 GOTO 10
```

Me despido esperando que estas rutinas sean útiles...

Francisco Capra C.

Estimado Francisco:

Te agradecemos realmente estas dos rutinas que nos enviaste. Realmente nos trae mucha satisfacción recibir las colaboraciones que TURBO news solicita a los lectores de la Revista.

Nos han gustado mucho los sonidos generados por tus rutinas y esperamos como tú que sean de utilidad para el resto de los atarianos y te invitamos para que sigas colaborando con nuestra tarea de difundir conocimientos acerca de el Atari... Nuestro amigo.

Marcelo Waldbaum.

TURBO NEWS:

En primer lugar, deseo felicitarlos por su Revista, es excelente.

Son varias las dudas que tengo:

-Cómo se puede hacer flashear una palabra en la pantalla?

-Cómo se pueden subrayar palabras en la pantalla?

-y por último tengo un diskette con el simulador de vuelo AT-FS2 y al ser cargado en mi computador Atari 130 XE, con una diskettera Atari XFF-551, me da errores durante la carga, a pesar de presionar la tecla "OPTION". Qué puede suceder?

Vicente Rios.

Estimado Vicente:

En primer lugar me gustaría agradecerte por ser un entusiasta lector de nuestra Revista TURBO news.

Existen varias formas para solucionar tus dos primeras interrogantes así que vamos a darte un ejemplo de una de ellas para que puedas incorporarlas en tus programas escritos en Basic.

Para hacer flashear una palabra en la pantalla, puedes imprimirla en video normal en una determinada posición y luego de un instante, imprimes la misma palabra pero en video inverso, siempre en la misma posición de tu monitor. Para escribir en video inverso, tu Atari posee en el extremo inferior derecho la tecla que convierte el video normal en inverso y viceversa.

Puedes incorporar en tus programas la siguiente rutina, que podrás utilizar con distintos mensajes para que se haga intermitente el mensaje por ti deseado. Fíjate que puedes definirle a la rutina la velocidad en la cual va a flashear tu mensaje, como así también la posición en la cual se va a ubicar dentro de la pantalla.

PROGRAMA PRINCIPAL

```
10 DIM A$(100),B$(100)
20 A$="ATARI"
30 POKE 752,1
40 A=10
50 B=10
60 DELAY=70
70 GOSUB 100
80 A$="TURBO NEWS TURBO NEWS"
90 GOSUB 100: END
```

SUBRUTINA PARA HACER FLASHEAR EN LA PANTALLA EL CONTENIDO DE LA VARIABLE A$.

VARIABLE A= COLUMNA
VARIABLE B= FILA
VARIABLE DELAY= VELOCIDAD

```
100 FOR I=1 TO LEN(A$)
110 B$(I,I)= CHR$(ASC(A$(I,I))+128)
120 NEXT I
130 POSITION A,B
140 PRINT A$
150 FOR J=1 TO DELAY
160 NEXT J
170 POSITION A,B
180 PRINT B$
190 FOR J=1 TO DELAY
200 NEXT J
210 IF PEEK(764)=255 THEN 130
211 POKE 764,255
220 RETURN
```

Como verás, la rutina hace intermitente el contenido de la variable A$ en las coordenadas A,B y con la velocidad definida en la variable DELAY. Lo hace hasta que tipées alguna tecla en tu Atari.

Para subrayar en la pantalla es más fácil, lo único que deberás hacer, es imprimir justo abajo de la palabra que desees subrayar un String que contenga tantos Control-M como letras tenga el mensaje que tú utilices. Esto es así, pues el caracter control-M fue diseñado especialmente para subrayar texto en tu pantalla.

Con respecto a tu diskette, existe la posibilidad que este se haya hechado a perder. Lo que te sugiero es que lo cargues en algún computador prestado, para determinar si existe algún problema con tu Unidad de Discos o realmente el disco se haya dañado.

LECCION 7

En esta ocasión vamos a enseñarles cómo el BASIC de nuestro computador ATARI permite a un programa modificarse a sí mismo. Analizaremos también, qué significa esto en términos prácticos.

Otra de las cosas que hacen del Basic Atari un lenguaje tan versátil, es que permite que un programa pueda modificarse a sí mismo. El Editor que posee, permite que se pueda operar con él, con otras fuentes además del teclado del computador. Dicho en otras palabras, pueden introducirse MDBOlíneas de programaMDNM no sólo tras escribirlas con el teclado, sino de la pantalla, colocadas allí por un programa de Basic, que esté funcionando en aquel momento. Explicaremos a continuación cómo debe hacerse.

Existe una posición de memoria que hace que el computador se ponga en modo RETURN. Este modo hace que, a partir de donde se encuentra el cursor, comience a oprimirse en forma continuada la tecla RETURN. Si en la pantalla por medio de instrucciones PRINT hemos desplegado una serie de instrucciones con su correspondiente número de línea, posicionando el cursor en la primera de ellas y poniendo el modo RETURN, éstas serán incorporadas al programa. Hay que tener mucho cuidado con los números de línea que van a tener esas nuevas instrucciones. Debemos, pues, fijarnos si ya existían instrucciones con dichos números.

Veamos cómo se lleva a la práctica:

Haremos un pequeño programa que agrega instrucciones REM al final del mismo.

```
10 GRAPHICS 0:POSITION 2,5
20 ? 1000;" REM ESTO ES UNA
PRUEBA"
30 ? 1010;" REM QUE AGREGA
INSTRUCCIONES"
40 ? 1020;" REM AL PROGRAMA"
50 ? "CONT"
60 POSITION 2,0
70 POKE 842,13:STOP
80 POKE 842,12
90 END
```

Analicemos las instrucciones:

La instrucción 10 limpia toda la pantalla y deja 4 reglones en blanco al principio. Veremos el motivo de esto en la explicación de la línea 70.

En las líneas 20 a 40, imprimimos en la pantalla instrucciones REM, incluidos

sus números de línea, poniendo especial cuidado en que dichos números no existan en la actualidad (1000, 1010, 1020).

En la instrucción 50, imprimimos la instrucción CONT, lo cual también veremos más adelante el por qué.

La instrucción 60 es utilizada para posicionar el cursor al comienzo de la pantalla.

En la instrucción 70 ponemos al computador en modo RETURN con la instrucción POKE 842,13. Luego paramos el programa ya que dicho modo funciona en el sistema inmediato de programación. Al detener el programa, el computador imprime automáticamente el mensaje "STOPPED AT LINE 70" en la posición 2.0 (al comienzo de la pantalla). Por eso si no hubiésemos dejado las cuatro líneas en blanco al principio, hubiese afectado las instrucciones impresas en la pantalla.

A partir de este momento, el computador comienza a hacer puros RETURNs, ingresando las instrucciones 1000, 1010, 1020 y CONT. Esta última instrucción va a hacer que vuelva a funcionar el programa, ejecutándose la instrucción siguiente que es POKE 842,12. Esta instrucción hace que el computador vuelva al modo normal y el programa continúe funcionando normalmente. En este caso particular, simplemente ejecutará la instrucción END. Si listamos el programa, veremos que existen tres nuevas instrucciones.

¿Para qué puede utilizarse todo esto? Te daremos dos aplicaciones prácticas, pero te garantizamos que con el tiempo, encontrarás varias más.

La primera de ellas, es usarla para borrar instrucciones de un programa. Muchos programas nos causan problemas de falta de memoria por su gran tamaño. Existen rutinas que serán utilizadas sólo una vez, así que una buena solución a la falta de espacio consistiría en borrarlas luego de usarlas. Por ejemplo, tenemos un vector que posee una agenda telefónica cuyos datos se encuentran almacenados en instrucciones DATA. La lectura de esas DATA se realiza una única vez al comienzo del programa y nunca más, pudiéndose borrar no sólo esas instrucciones, sino también la rutina que permitió cargar el vector.

Veamos cómo se haría con un ejemplo sencillo:

```
10 GRAPHICS 0:POSITION 2,4
20 PRINT 100:PRINT 110:PRINT
120
30 PRINT "CONT"
40 POSITION 2,0
50 POKE 842,13: STOP
60 POKE 842,12
70 END
100 REM LINEAS
110 REM PARA
120 REM BORRAR
```

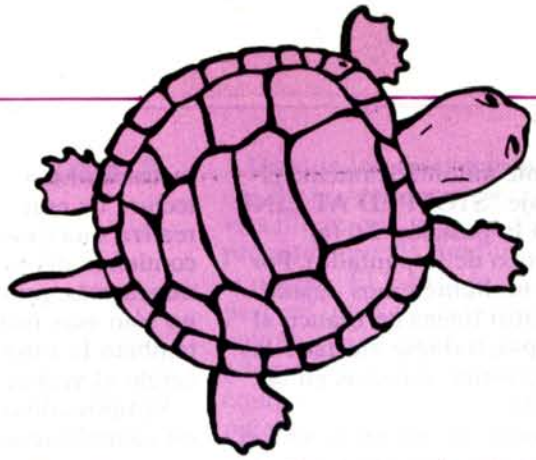
Luego de ejecutarse el programa, al listarlo, veremos cómo las instrucciones 100,110 y 120 han desaparecido.

Otra aplicación es agregarle a un programa instrucciones DATA que contengan cualquier tipo de información. Vamos a hacer un pequeño programa que agregue instrucciones DATA con nombre de personas.

```
1 DIM NOM$ (20)
10 GRAPGHICS 0
20 PRINT "INGRESE NOMBRE:";:
INPUT NOM$
30 PRINT "INGRESE NUMERO DE
LINEA DEL LISTADO:";: INPUT N
40 GRAPHICS 0:POSITION 2,4
50 ? N; "DATA"; NOM$
60 ? "CONT"
70 POSITION 2,0
80 POKE 842,13: STOP
90 POKE 842,12: GOTO 10
```

La instrucción 30 nos pide que le indiquemos el número de línea donde deseamos poner la nueva instrucción DATA creada.

Te desafiamos a que le encuentres nuevas aplicaciones y nos escribas los resultados de tus investigaciones. Hasta pronto!!



En este número aprenderemos a guardar las instrucciones creadas por nosotros para ser usadas en cualquier momento.

APRENDIENDO A DIBUJAR CON LOGO

LECCION 5

Cuando uno le va enseñando instrucciones al computador, éste las va guardando en un área de trabajo de manera de poder usarlas más adelante, cuando se lo solicitemos. Sin embargo, si se apaga el computador, todas las instrucciones allí guardadas son olvidadas. Para no perder toda esa información en el momento de apagar el computador, es que existen los medios de almacenamiento, es decir, la Cassettera o la Diskettera. También puede utilizarse la Impresora para guardar en papel las instrucciones creadas. Comenzaremos viendo el uso de la Diskettera.

DISKETTE

Toda la información guardada en un diskette necesita tener un Nombre que la diferencie del resto. ATARI tiene ciertas reglas para la creación de dichos nombres. Estos pueden tener ocho caracteres (letras o números) o menos, más una extensión optativa de otros tres caracteres. El primer caracter del nombre debe ser obligatoriamente una letra y todas las letras deben estar en mayúscula. La extensión está separada del resto por un punto y se utiliza para identificar mejor un archivo. Por ejemplo, puede usarse para saber quién fue el creador:

JUEGO.MAU, si lo hizo Mauro
JUEGO.ROB, si lo hizo Roberto

Puede usarse para saber cuántos archivos se lleva:

JUEGO.001
JUEGO.002

Veamos ahora cuáles son las instrucciones necesarias:

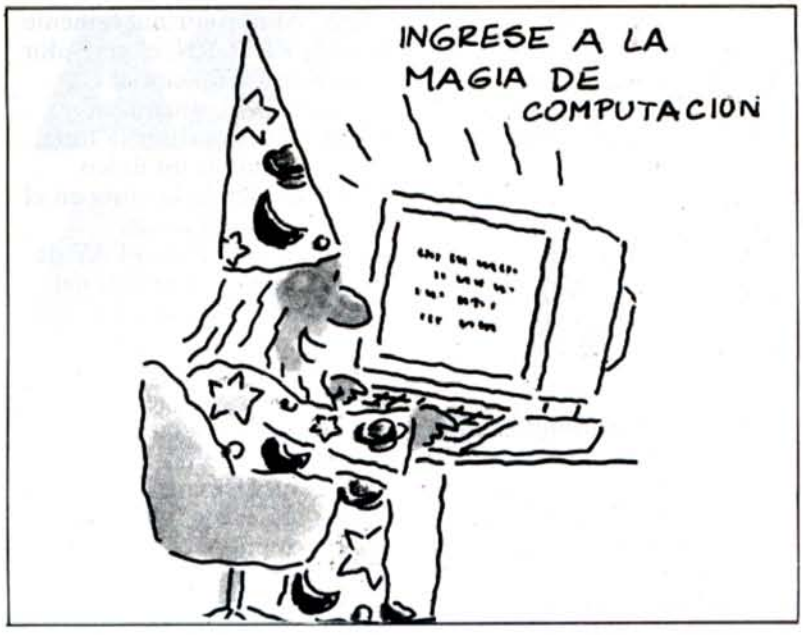
INSTRUCCION GRA "Dn: NOMBRE. EXTENSION

El significado es el siguiente: Grabar en la diskettera n, todas las instrucciones creadas hasta el momento, con el siguiente nombre y extensión. Por ejemplo:

GRA "D1:JUEGO.1.
GRA "D2:JUEGO.2.

Esta instrucción puede usarse en el caso de tener más de una diskettera.

Te recordamos que para usar la diskettera, hay que encenderla antes que el computador y ponerle dentro cualquier disco que contenga



un DOS (Sistema Operativo de Disco) al hacerlo.

Para poder recuperarlas, se utiliza la siguiente instrucción:

INSTRUCCION CARG "Dn:NOMBRE. EXTENSION

El significado es el siguiente: Cargar el computador desde la diskettera n, con todas las instrucciones que se encuentren en el archivo NOMBRE.EXTENSION. El computador, a medida que lo va haciendo, indicará cuáles son las instrucciones que tenía guardadas en dicho archivo.

Para probar estas dos instrucciones, utilizaremos nuestras viejas y conocidas instrucciones RECTANGULO y CUADRADO. Primero debemos ingresarlas:

```
CR CUADRADO :LADO
REPITE 4 [AV :LADO DE 90]
FIN

CR RECTANGULO :LADO1
:LADO2
REPITE 2 [AV :LADO1 DE 90 AV
:LADO2 DE 90]
FIN
```

A continuación, vamos a guardar estas instrucciones con el nombre PRUEBA.001

```
GRA "D: PRUEBA.001
```

Cuando se utiliza la diskettera 1 para almacenar o grabar, se puede no poner el número 1 al lado de la letra D.

Para comprobar que hemos hecho las cosas correctamente, borremos todas las instrucciones creadas (RECTANGULO Y CUADRADO) con la instrucción ELTOD (eliminar todo) y luego utilicemos la instrucción para cargar:

```
ELTOD
CARG "D: PRUEBA.001
```

El computador contestará:

```
CUADRADO ANOTADO
RECTANGULO ANOTADO
```

Y podremos usarlas. Compruébalo haciendo CUADRADO 30 y RECTANGULO 60 50

Otras instrucciones que existen para el uso de diskettera son: CATA y DES-

INSTRUCCION CATA "Dn:

Su significado es el siguiente: "Muéstrenos el catálogo de archivos que existe en el disco que se encuentra en la diskettera n". Por ejemplo:

```
CATA "D:
```

Luego de ejecutada esta instrucción, el computador nos mostrará todos los archivos que se encuentran en el disco colocado en la diskettera 1. Esta instrucción es muy útil cuando deseamos cargar un archivo y no nos acordamos del nombre que posee.

INSTRUCCION DES "Dn: NOMBRE. EXTENSION

Su significado es el siguiente: Destruya el archivo NOMBRE.EXTENSION que se encuentra en la diskettera n. Se utiliza cuando no deseamos ocupar más un archivo, y está usando un lugar innecesario dentro del diskette.

A continuación veremos el uso de la Cassettera:

Las diferencias que existen en el uso con respecto a la diskettera son las siguientes: El grabador no puede buscar automáticamente un programa dentro de un cassette como lo hace la diskettera. Lo debes ubicar tú en forma manual. Para ello, la cassettera posee un cuenta-vueltas que te permite anotar en qué número

de vuelta comienza cada archivo y así poder luego recuperarlo. Por esto mismo los archivos de cassette no necesitan tener nombre, ya que la búsqueda la realiza uno mismo.

Las instrucciones son las mismas, salvo la instrucción DES que no existe en cassette. Para indicarle al computador que queremos utilizar una cassettera, se utiliza la letra C. Así, las instrucciones que resultan son:

GRA "C:
CARG "C:
CATA "C:

Recuerda que siempre, antes de ejecutar estas instrucciones, deberás posicionar la cinta en el lugar correspondiente, con la ayuda del cuenta-vueltas. Para borrar un archivo, basta con grabarle algo encima. Por eso, la instrucción DES no tiene sentido.

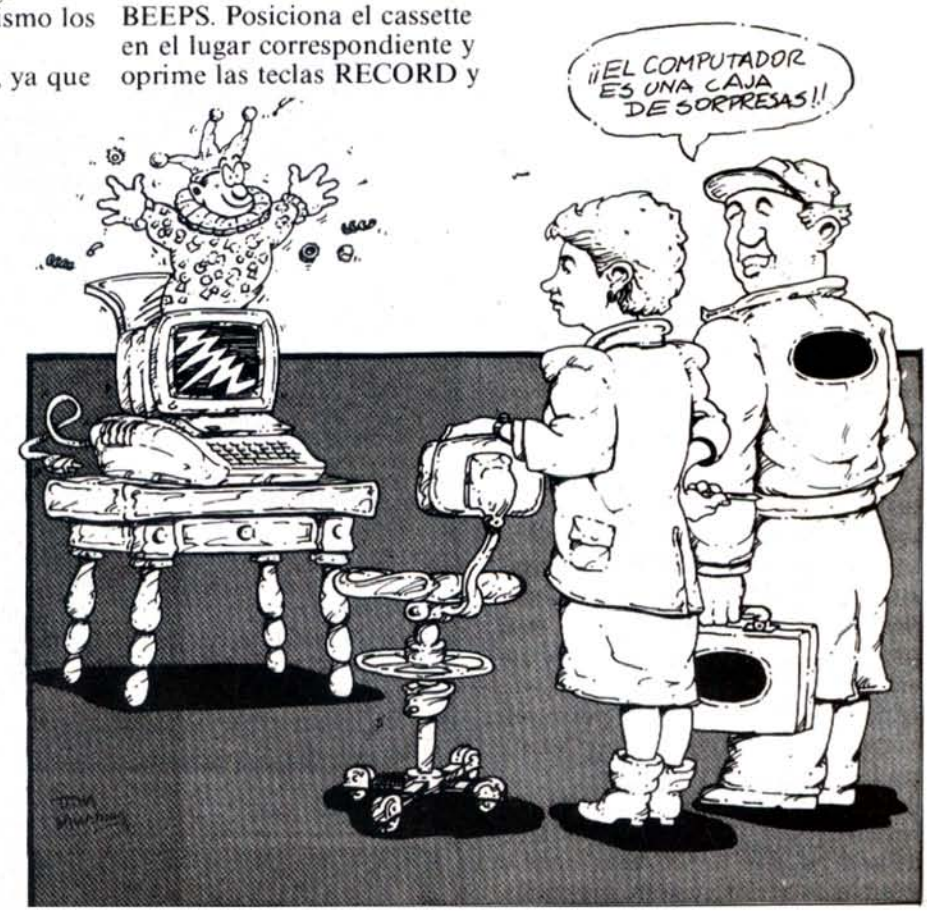
Al escribir la instrucción GRA "C: y oprimir la tecla RETURN, escucharás dos BEEPS. Posiciona el cassette en el lugar correspondiente y oprime las teclas RECORD y PLAY. Al oprimir nuevamente la tecla RETURN, el grabador comenzará a funcionar.

Al escribir la instrucción CARG "C: y oprimir la tecla RETURN, oirás un único BEEP. Poniendo la cinta en el lugar correspondiente, oprimiendo la tecla PLAY de tu grabador y RETURN del computador, el grabador comenzará a funcionar.

Para usar la instrucción CATA deberás seguir los mismos pasos que con la instrucción CARG.

Finalmente para el uso de la impresora existe una única instrucción que es GRA "P: Una vez ejecutada esta instrucción, todos los comandos que hayamos creado, serán impresos en papel.

Nos vemos en el siguiente número!!!.



ASSEMBLER

ASSEMBLER

LECCION 7

En las lecciones anteriores, hemos analizado la mayoría de las instrucciones necesarias para programar en Assembler, pero a partir de esta lección cambiaremos, espero que para bien, la mecánica del aprendizaje de este apasionante lenguaje.

Comenzaremos a consolidar el manejo de todas las instrucciones del 6502 en un programa complejo que a medida que vayamos completando las lecciones de este curso iremos analizando rutina a rutina.

Hace algún tiempo atrás un amigo: Roberto Domingues, me mostró un cartel luminoso pasamensajes que había desarrollado él para efectos publicitarios. Estaba diseñado sobre la base de un procesador por todos nosotros conocido "El 6502" y contaba con una pantalla de 32 columnas de 7 leds cada una, las cuales al encenderse y apagarse generaban un efecto muy atractivo para ser utilizado en vitrinas comerciales.

Lo interesante de este equipo, además de su capacidad gráfica, era que no necesitaba de un costoso teclado para su programación, como el resto de los equipos similares que existen en el mercado. Trabaja conjuntamente con cualquier computador, el cual es utilizado para enviarle su programa a través de una línea de comunicación Standard RS-232. El equipo cuenta con una gran variedad de efectos gráficos, los cuales se pueden generar a partir de instrucciones que en el cartel están representadas por la recepción desde el computador de determinados bytes.

En esta sección del curso de Assembler comenzaremos a desarrollar por fin nuestro primer sistema totalmente programado en lenguaje de máquina.

El problema que el cartel tenía, era que el usuario se veía en la necesidad de programar el cartel junto con su manual de instrucciones, pues estos códigos eran muy difíciles de recordar.

Para solucionar este inconveniente de Software decidimos desarrollar un Lenguaje de Programación especial para ser utilizado con el cartel Lixxon DMP-32. Por supuesto que lo desarrollamos con Atari, pues en el mercado de los computadores hogareños es el líder indiscutido.

Después de unos cuantos meses el programa estaba terminado y el usuario del cartel ya programaba como en Basic los efectos de gráficos que quería que su Lixxon realizara.

Recorriendo mi caja porta-diskettes y haciendo un estudio detallado de todos los programas desarrollados en Assembler, pensé que este era el más adecuado para

ASSEMBLER

Continuemos introduciéndonos más en tu computador

incorporar en el curso, así que acá vamos!!!

El lenguaje que desarrollamos era similar al editor del MAC/65 o bien el Turbo Basic, pues contaba con la posibilidad de numerar automáticamente las instrucciones, hacer renumeraciones de línea sin necesidad de contar con algún utilitario y así mismo permite borrar del listado determinadas instrucciones.

En esta edición de TURBO news, comenzaremos por describir la estructura general del sistema mediante su diagrama de flujo e incorporamos en la sección de programas la primera parte del listado del programa. En las siguientes secciones del curso de Assembler analizaremos las distintas rutinas que conforman el diagrama de flujo y seguiremos listando el programa.

El diagrama de flujo del programa está compuesto por un ciclo permanente, en el cual el programa se queda esperando que el usuario por algún motivo presione la tecla Return. Cuando lo hace, el lenguaje lixxon interpreta que una línea de programa o bien un comando fue ingresada al listado. Si fue una línea de programa como ser:

100 VELOCIDAD 1O

Lo primero que este compilador realiza es una eliminación de los posibles espacios en blanco dejados por el usuario a la izquierda del número de la instrucción. Luego se procesa comprimiendo su información para luego almacenarla en forma consecutiva en la tabla de instrucciones que el sistema tiene definida en Ram. Si en vez de una instrucción fue un comando de la lista de comandos permitidos en el editor se ejecuta el mismo según la rutina que lo desarrolla y el sistema vuelve a solicitar al usuario una nueva línea en el programa.

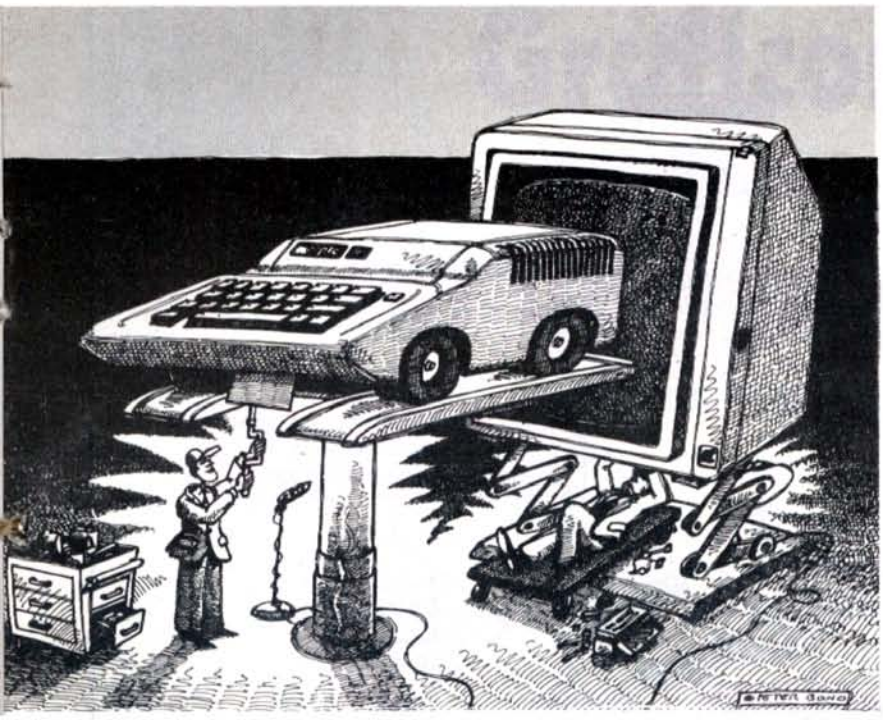
Los comandos que este sistema tiene incorporados son los que podemos encontrar en la siguiente lista:

COMANDO	ABREVIATURA	ACCION DESARROLLADA
CARGAR	C	Carga un programa desde la cassettera o diskettera.
GRABAR	G	Graba un programa en la cassettera o diskettera.
LISTAR	L	Lista el programa en memoria.
NUEVO	---	Borra el programa de la memoria.
RENUMERAR	R	Renumera un programa.
NUMERAR	NUM	Numera instrucciones automáticamente.
BORRAR	B	Borra determinadas líneas del programa.
ENVIAR	E	Compila el programa y se lo envía al cartel.
DOS	D	Llama al DOS.

Cada vez que el sistema reconoce la existencia de un comando al encontrar que la línea no tiene número de instrucción, se recorre una tabla de definición de comandos verificándose que el comando en su nombre es válido. Si no lo es se imprime en la pantalla el mensaje de error por comando inexistente, pero si lo es, en dicha tabla se encuentra la dirección de la subrutina que ejecutara el comando para luego volver al ciclo normal del programa.

Cuando el comando ingresado fue el de numeración automática, el sistema define a la variable FLAGNUM como distinta de cero, para saber en futuras instrucciones que debe generar automáticamente el número de instrucción.

El diagrama de flujo parte en su inicio definiendo a esta variable FLAGNUM en cero, asumiendo que, al partir, el programa no numera automáticamente. Luego define



todos los punteros necesarios para administrar la zona de memoria en la cual se irán almacenando las instrucciones.

Una vez realizada esta tarea, verifica si el comando de numeración automática, dejó en un valor distinto de cero a la variable Flagnum. Si es así, el diagrama genera e imprime en la pantalla el siguiente número de instrucción. Luego le pide al usuario que ingrese una nueva línea y si esta es RETURN, coloca a Flagnum en cero, pues el usuario quiso eliminar la numeración automática.

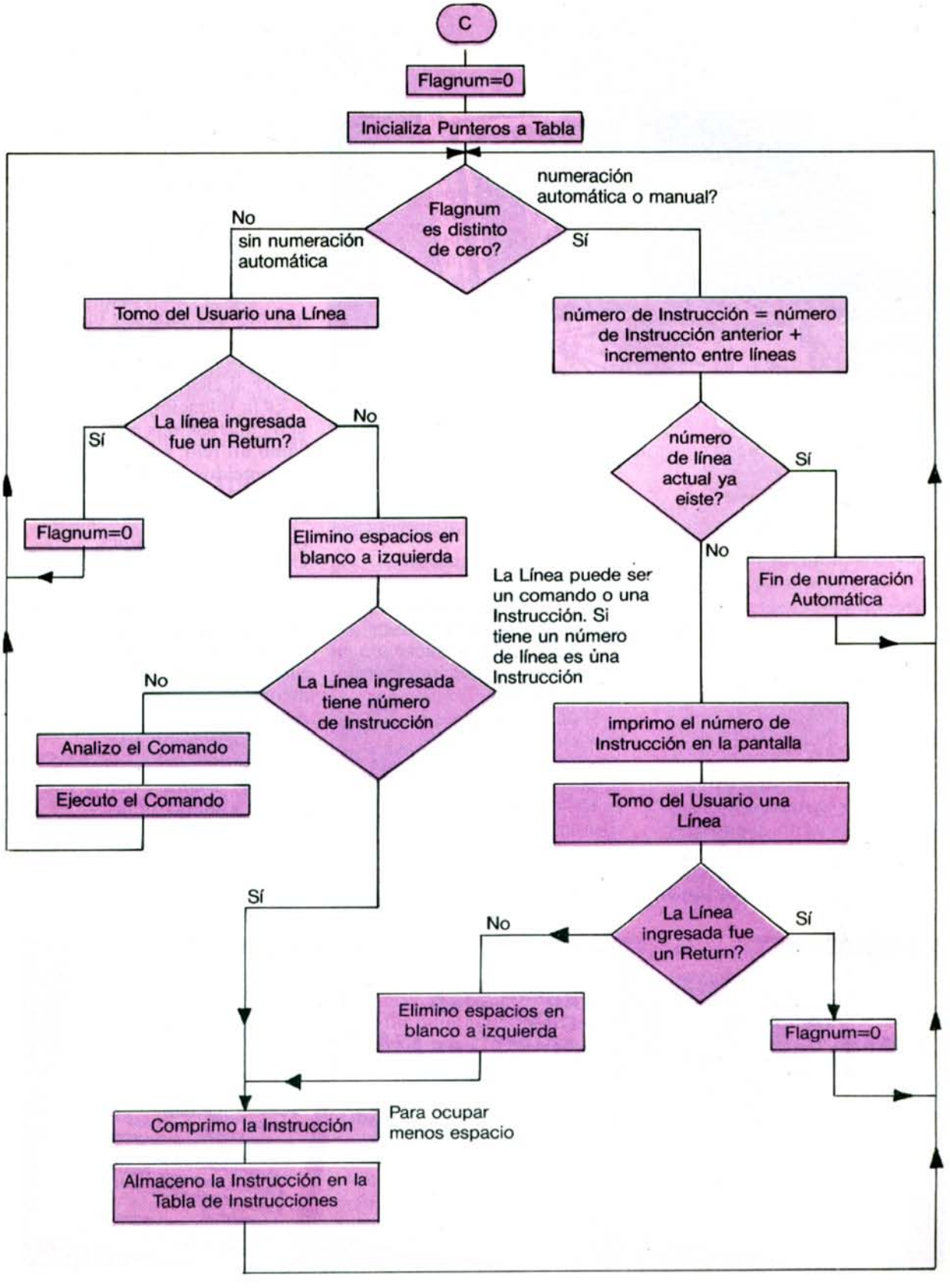
Si la línea ingresada no fue un Return, se comprime la instrucción y es almacenada en la tabla de instrucciones para luego continuar en el ciclo.

Si el programa no se encuentra numerando, automáticamente se le pide al usuario que ingrese la línea. Si es RETURN, al igual que en el caso anterior se sigue en el ciclo, pero si la línea no es un Return, ésta es analizada para determinar si es una instrucción para el lenguaje o bien un comando para el computador. Esto se determina sabiendo si el primer caracter de la línea es un número. Si lo es, estamos ante una instrucción del lenguaje que como tal está compuesta por un número de instrucción y el código que la conforma.

Si el primer caracter de la línea no es un número, estamos recibiendo un comando de parte del usuario que hay que analizar para ver si es válido en su sintaxis, ejecutarlo y si corresponde volver al ciclo del programa.

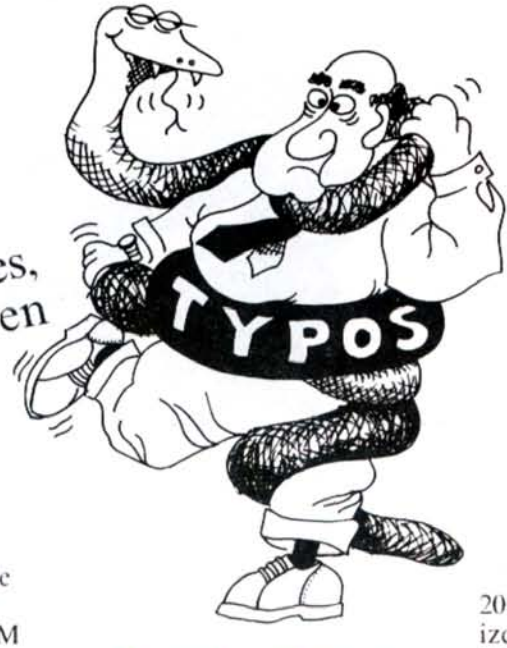
Es la intención de TURBO news que asumas este proyecto como un reto, pues estamos convencidos que al terminar de analizarlo, no tendrás límites en cuanto a la programación en lenguaje Assembler. Comenzamos con la estructura general del sistema y su diagrama de flujo, para que en las siguientes ediciones de nuestra revista analicemos cada una de las rutinas que incorporamos en el Sistema. Asi mismo te solicitamos que vayas ingresando este programa en tu computador, pues es bastante extenso y lo iremos listando en las sucesivas revistas.

Hasta pronto!!!


C
Flagnum=0
Inicializa Punteros a Tabla
Flagnum es distinto de cero?
numeración automática o manual?
No
sin numeración automática
Sí
Tomo del Usuario una Línea
número de Instrucción = número de Instrucción anterior + incremento entre líneas
La línea ingresada fue un Return?
Sí
No
Flagnum=0
Elimino espacios en blanco a izquierda
número de línea actual ya eiste?
Sí
No
Fin de numeración Automática
La Línea puede ser un comando o una Instrucción. Si tiene un número de línea es úna Instrucción
La Línea ingresada tiene número de Instrucción
No
Analizo el Comando
Ejecuto el Comando
imprimo el número de Instrucción en la pantalla
Tomo del Usuario una Línea
Sí
La Línea ingresada fue un Return?
No
Sí
Elimino espacios en blanco a izquierda
Flagnum=0
Comprimo la Instrucción
Para ocupar menos espacio
Almaceno la Instrucción en la Tabla de Instrucciones

Gráficos por computadora

SEGUNDA PARTE



No te enredes, los gráficos en Atari son sencillos

En este número veremos cómo se realizan los movimientos de los gráficos de PLAYER-MISSILE, tanto horizontal como vertical y diagonal.

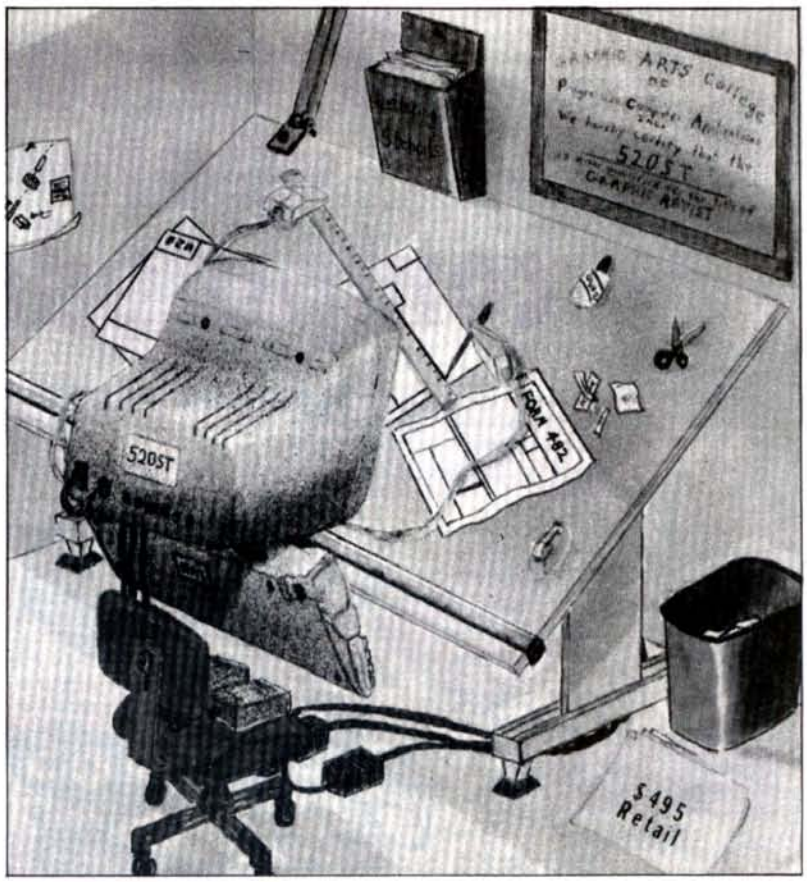
Cada Player o misil, puede desplazarse en forma MDBOindependiente.MDNM ya sea en sentido horizontal, vertical o diagonal.

El movimiento horizontal, de izquierda a derecha y viceversa, es el más sencillo de todos, ya que cada player, al igual que los misiles, posee una posición de memoria que le indica al computador cuál es su ubicación horizontal. Para modificar la misma en la pantalla, basta con modificar dicho valor. Como todas las posiciones de memoria, los valores de ésta varían de 0 a 255. Sin embargo, sólo un grupo de esos valores hacen que sea visible, esto dependiendo de cada aparato de televisión.

Normalmente los valores del 20 al 60 marcan el margen izquierdo y los mayores a 200 a 245 el derecho. Una consecuencia de esto es que la mejor manera de hacer desaparecer un Player-Missile es dándole un valor de 0 a la posición de memoria correspondiente. Los valores de las posiciones de memoria anteriormente nombrados son los siguientes:

PLAYER 0 53248
PLAYER 1 53249
PLAYER 2 53250
PLAYER 3 53251
MISSILE 0 53252
MISSILE 1 53253
MISSILE 2 53254
MISSILE 3 53255

Gráficos por computadora



Estas posiciones de memoria forman parte de los denominados REGISTROS DE HARDWARE y reciben esa denominación pues se ocupan para que el programador pueda transmitirle datos al computador. Dicho de otra manera, sólo son de escritura, utilizando para ello la instrucción POKE. La lectura de estos registros con la instrucción PEEK, no entrega el valor de la posición horizontal. Veremos más adelante qué significa dicho valor. Para probar lo anteriormente mencionado hagamos la siguiente prueba:

```
POKE 53248,60
PRINT PEEK (53248)
```

Y el computador imprimirá el valor 0 y no 60 como sería si no fuera un registro de hardware.

El desplazamiento vertical es un poco más difícil de realizar, ya que no existe ningún registro de posición vertical. Lo que debemos hacer es copiar todos los bytes que componen nuestro dibujo, dentro de la zona de memoria que le corresponda a cada PLAYER-MISSILE, a la nueva ubicación que deseemos poniendo cuidado de borrar el lugar donde se encontraba anteriormente. Para que el movimiento sea fluido y continuo como lo era el horizontal y no vaya temblando, el lenguaje Basic resulta muy lento, por lo que conviene hacerlo en lenguaje de máquina. No se asusten, les daremos a continuación un truco que les permitirá hacerlo muy fácilmente. Este se basa en la utilización de las cadenas alfanuméricas, por lo que le daremos una breve introducción de ellas antes de entrar de lleno en el tema.

CADENAS ALFANUMERICAS.

Cada vez que se introduce una línea en Basic que mencione a una nueva variable alfanumérica, cargan sus valores en dos tablas especiales que el Basic posee para su reconocimiento. La primera, llamada tabla de nombres de variables y la segunda denominada tabla de valores de variables (VVTP). La ubicación exacta de esta segunda tabla se encuentra almacenada en las posiciones 134 y 135 del computador y para saber cuál es, debemos hacer la siguiente operación matemática:

```
VVTP=PEEK(134)+PEEK(135)*256
```

Los contenidos de las variables se encuentran grabados en la memoria, a partir de una posición llamada STARP y que se puede saber con la siguiente fórmula matemática:

```
STARP=PEEK(140)+PEEK(141)*
256
```

En la VVTP, por cada variable, se utilizan ocho bytes con la siguiente información:

byte 1= Valor que indica si ya ha sido dimensionada la variable. 128 significa no dimensionada y 129 sí dimensionada.

byte 2= Posición dentro de la tabla de nombres de variables. Depende del orden cronológico en que fueron introducidas al programa y no al orden en que se encuentran en el mismo.

bytes 3 y 4= Indican el desplazamiento con respecto al STARP en que comienza cada variable. La información se encuentra almacenada en dos bytes o sea que, para saber cuál es el desplazamiento exacto, debemos hacer el PEEK del primero sumado al PEEK del segundo multiplicado por 256.

bytes 5 y 6= Indican la longitud actual ocupada por dicha variable (la que devolvería de aplicársele la instrucción LEN) y para saber su valor debe hacerse igual que en el caso anterior.

bytes 7 y 8= Indican cuál es la longitud máxima que puede tener y es la que se le asigna con la instrucción DIM.

Veamos en un ejemplo, cómo resulta todo esto:

```
10 DIM A$(20), B$(261)
20 A$= "MAURO"
30 B$= "MARIANA"
```

La tabla VVTP va a quedar de la siguiente manera:

	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8
A$	129	0	0	0	5	0	20	0
B$	129	1	20	0	7	0	5	1

El secreto de todo el truco consiste en modificar los bytes 3 y 4 de las variables de manera que la zona de memoria ocupada por éstas coincida con las zonas de memoria que utilizan los Player-Missiles. De esta manera, cualquier cosa que hagamos con dichas variables hará que influya directamente sobre los Player-Missiles. En otras palabras, para realizar cualquier movimiento vertical, basta con hacer una asignación a la variable correspondiente, que es lo mismo que hacer un movimiento de bytes a velocidad de lenguaje de máquina.

Veamos todo esto en un ejemplo:

Para mover un Player de un lugar a otro de la pantalla en sentido vertical utilizaremos cuatro variables alfanuméricas:

BLANCO$= Variable de 256 caracteres de largo y toda llena de espacios en blanco.

DIBUJO$= Variable cuya longitud depende del dibujo del Player que hayamos hecho y que vamos a suponer de longitud 12.

AUX$= Variable auxiliar de longitud 256.

MISSILE$= Variable de longitud 256 y cuyos valores 3 y 4 de la VVTP apuntan a la zona de memoria ocupada por el PLAYER 1.

El movimiento consta de tres pasos.

1- Blanqueamos la variable auxiliar:

```
AUX$=BLANCO$
```

2- Ponemos el dibujo en la posición vertical que deseamos dentro de la variable auxiliar, llamando a dicha posición YPOS:-

```
AUX$(YPOS)=DIBUJO$
```

3- Ponemos el resultado de estas operaciones en la variable que coincide con el PLAYER 1 (PLAYER$)

```
PLAYER$=AUX$
```

IMACO
M R
El centro electrónico del centro de Santiago
ESTADO 46 - FONOS: 392835 - 394231

Gráficos por computadora

Nótese que el borrado del Player de su posición anterior se realiza automáticamente en el paso 1 cuando a AUX$ le asignamos puros blancos, por más que lo mismo se concrete recién en el paso 3.

Si deseamos que el movimiento sea de un único paso hacia arriba o hacia abajo para simular un movimiento continuo, podemos saltear los pasos 1 y 3, pero agregando a la variable DIBUJO$ dos bytes en blanco, uno al comienzo y otro al final, para que al mover de uno en uno el dibujo, automáticamente se vaya borrando el dibujo antiguo. El movimiento quedaría de la siguiente manera:

Para mover el dibujo una posición hacia arriba:

```
PLAYER$(YPOS-1)=DIBUJO$
```

Dibujo$ tiene ahora una longitud de 14

Para moverlo una posición hacia abajo:

```
PLAYER$(YPOS+1)=DIBUJO$
```

Para obtener movimientos diagonales, basta con combinar los movimientos vertical y horizontal. Puede hacerse en 45 grados combinando uno y uno o de cualquier otra forma dependiendo de la combinación dada.

Hasta aquí llegamos en este número. Nos vemos en el siguiente!!

EXPLORANDO EL DISKETTE

En este número describiremos uno de los soportes de información de uso más generalizado: el Diskette.

Comenzaremos por estudiar físicamente al diskette. Este está constituido por un soporte flexible recubierto por una capa de material magnético. La presencia de magnetización indica el símbolo 1 y su ausencia el número 0. Estos se encuentran contenidos en una envoltura de protección de forma rectangular, como se ve en el dibujo más abajo.

Dicha envoltura posee una ranura por la cual se desliza la cabeza lectora hacia adelante y atrás. Como el disco se encuentra en continuo movimiento de rotación, cada parte de éste puede ser alcanzado por la cabeza. Esta es una de las grandes ventajas que posee el disco frente al cassette. Para llegar a cualquier punto, le basta con girar un poco y posicionar la cabeza en el lugar exacto. En el caso de la cinta, la misma es recorrida secuencialmente, o sea que para acceder a cualquier punto, debe primero haber pasado por todos los puntos anteriores.

Los discos pueden ser grabados en una sola cara o por ambas, representado esto en los discos con el símbolo 1S o 2S. La letra S es inicial de la palabra en inglés SIDE (Lado). Cuando los discos vienen marcados como 1S, no significa que no se puedan utilizar por su cara B, sino simplemente que el fabricante no garantiza su buen funcionamiento, cosa que frecuentemente ocurre. Otra característica que poseen los discos es la densidad de las zonas de magnetización. De ella depende que se pueda en un mismo disco almacenar mayor o menor información. Esto viene representado con el símbolo 1D para Simple Densidad y 2D para Doble. Ocurre para 1D lo mismo que en el caso de 1S, el fabricante no garantiza su buen uso, pero uno puede utilizarlo en 2D.

La cubierta protectora posee también una muesca en su parte superior derecha que es utilizada como protección de escritura. Su funcionamiento es similar a la muesca que poseen en su parte trasera todos los cassettes. Cuando ésta se encuentra al descubierto puede grabarse y leerse del disco. Si es tapada puede leerse pero no grabarse.

Generalmente los discos no poseen dicha muesca en la parte superior izquierda. Para grabar entonces su segunda cara debemos hacer la muesca nosotros mismos. Existe un aparatito manual que lo efectúa en el lugar exacto. Sin embargo, podemos con la ayuda de una afilada tijera o cortante, simular la muesca poniendo mucho cuidado de hacerlo en el lugar y con las dimensiones correctas.

Todos los discos pueden ser utilizados con cualquier computador. Cuando ponemos un disco en nuestro computador, lo primero que debemos hacer es Formatearlo. En esta etapa, cada computador "adapta" el diskette a su sistema de ordenamiento de la información. El orden se efectúa en base a pistas y sectores. Pistas son círculos concéntricos totalmente independientes entre sí (a diferencia de un disco musical donde todos los círculos están unidos formando un único zurco). Cuando la cabeza se desplaza hacia adelante y hacia atrás, lo que hace es recorrer una pista a la otra. Los sectores son divisiones dentro de cada pista, cada uno de los cuales tiene un número determinado de bytes de información y control. Se establece así, un mapa del disco donde cada zona específica puede ser direccionada mediante un número de pistas y un número de sector. La dirección mínima que se puede dar es la de un sector. No se pueden seleccionar partes de éste, por lo que la escritura y lectura de datos se efectúa por bloques cuya longitud mínima es un sector. Lo anteriormente mencionado es totalmente transparente para el usuario. Cada vez que solicitamos

EL DISKETTE

algún dato, el computador se encarga de leer todos los sectores que contienen el dato solicitado y lo recupera.

ESTRUCTURA FISICA DE UN DISKETTE

SENTIDO DE GIRO

ORIFICIO DE ARRASTRE

CABEZA MAGNETICA

DISCO EN SOLO LECTURA

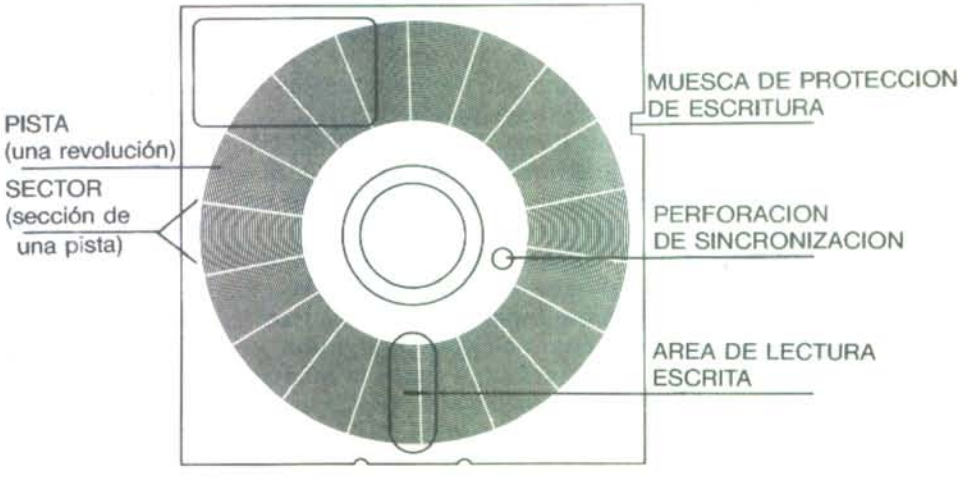
DISCO TAMBIEN EN ESCRITURA

MOVIMIENTO DE LA CABEZA A LO LARGO DE UNA GENERATRIZ

PROTECCION

VENTANILLA DE ACCESO PARA LA CABEZA

DISKETTE FORMATEADO



Los discos pueden variar en cuanto a la cantidad de pistas, sectores y bytes que cada uno de ellos poseen. Dependen de cómo se les haya formateado.

Atari posee tres tipos de formatos (teniendo en cuenta sólo una cara del disco). Estos son: SIMPLE, ENHACED 0 1050 y DOBLE. Veamos las características de cada uno de ellos.

	SIMPLE	1050	DOBLE
PISTAS	40	40	40
SECTORES/PISTA	18	26	18
BYTES/SECTOR	128	128	256
TOTAL SECTORES	720	1040	720
TOTAL BYTES	92160	133120	184320

Veamos cómo están constituidos lógicamente los discos en su interior. Para administrar los discos, ATARI guarda para su uso exclusivo una serie de sectores y algunos bytes dentro de cada sector que le sirven para enlazar los sectores que pertenecen a un mismo archivo. Debido a que éstos pueden modificarse, agrandarse o achicarse en cualquier momento, puede ser que físicamente los sectores de un mismo archivo no se encuentren en forma correlativa. Por eso es que existen los bytes de enlace. Esto hace que un administrador de disco, relativamente pequeño, pueda manejar la información que se encuentra en los discos en forma dinámica. La principal desventaja es que en este esquema no se puede acceder en forma directa a un sector intermedio de algún archivo, sino que recorrer secuencialmente todo el archivo para llegar al punto deseado. tro problema es que, de producirse un error en uno de los sectores intermedios, toda la información a continuación se pierde.

Esto no implica que no puedan hacerse archivos de acceso directo. Para hacerlo, podemos manejar los sectores nosotros mismos con las rutinas de CIOV dadas en números anteriores, evitando el administrador de diskette.

Los sectores dentro de un disco puden dividirse en tres grupos:

-VOLUME TABLE OF CONTENTS (VTOC).

Es un único sector que indica qué sectores del disco están ocupados y cuáles libres.

-DIRECTORIO.

Un grupo de ocho sectores puestos en forma consecutiva, utilizados para almacenar los nombres de los archivos, en qué sector comienza y otros datos de interés.

-SECTORES DE INFORMACION.

Sectores conteniendo información almacenada por el usuario más los bytes de enlace.

Veamos cada uno de estos sectores con mayor detalle.

-VTOC

Se encuentra en el sector 360. Su configuración interna es la siguiente:

BYTE INICIAL DEL CAMPO	LONGITUD DEL CAMPO	DESCRIPCION
0	1	Reservado
1	1	Número total de sectores
3	2	Número de sectores vacíos
5	5	Reservados
10	90	Mapa de sectores vacíos
100	28	Utilizados si se ocupan más de 720 sectores

EL DISKETTE

La parte más importante son los 90 bytes del mapa de sectores vacíos. Como cada byte tiene 8 bits, la cantidad de bits disponibles son 90*8=720, lo cual equivale a un bit por sector. Un bit puesto en 0 significa que ese sector está siendo usado. Un bit puesto en 1 significa que está desocupado.

Los bytes del 100 en adelante se utilizan cuando la densidad del formateo no es simple.

-DIRECTORIO DEL DISCO

El Directorio comienza en el sector 361 y continúa por ocho sectores contiguos hasta el 370. Estos sectores fueron escogidos porque se encuentran en el centro del disco y tienen el menor tiempo promedio de acceso desde cualquier lugar del disco. Cada sector del directorio tiene capacidad para almacenar 8 archivos. Por ello, en los 8 sectores podemos almacenar hasta 64 archivos. Cada uno de ellos ocupa 16 bytes, como está ilustrado en la figura a continuación:

El primer byte es utilizado para brindar información de cuál es el estado de la presente operación.

Los bytes 1 y 2 representan en LSB y MSB (byte de bajo y alto orden), la cantidad de sectores que ocupa el archivo.

Los bytes 3 y 4 representan en LSB y MSB, el número de sectores donde comienza el archivo.

Los bytes 5 al 12 tienen el nombre del archivo.

Los bytes 13 al 15 tienen el nombre de la extensión del archivo. Por ejemplo: .BAS o .ASM.

BYTE INICIAL DEL CAMPO	LONGITUD DEL CAMPO	DESCRIPCION
0	1	Byte de señal
1	2	Cantidad de sectores del archivo
3	2	Número de sectores inicial
5	8	Nombre del archivo
10	90	Nombre de la extensión del archivo

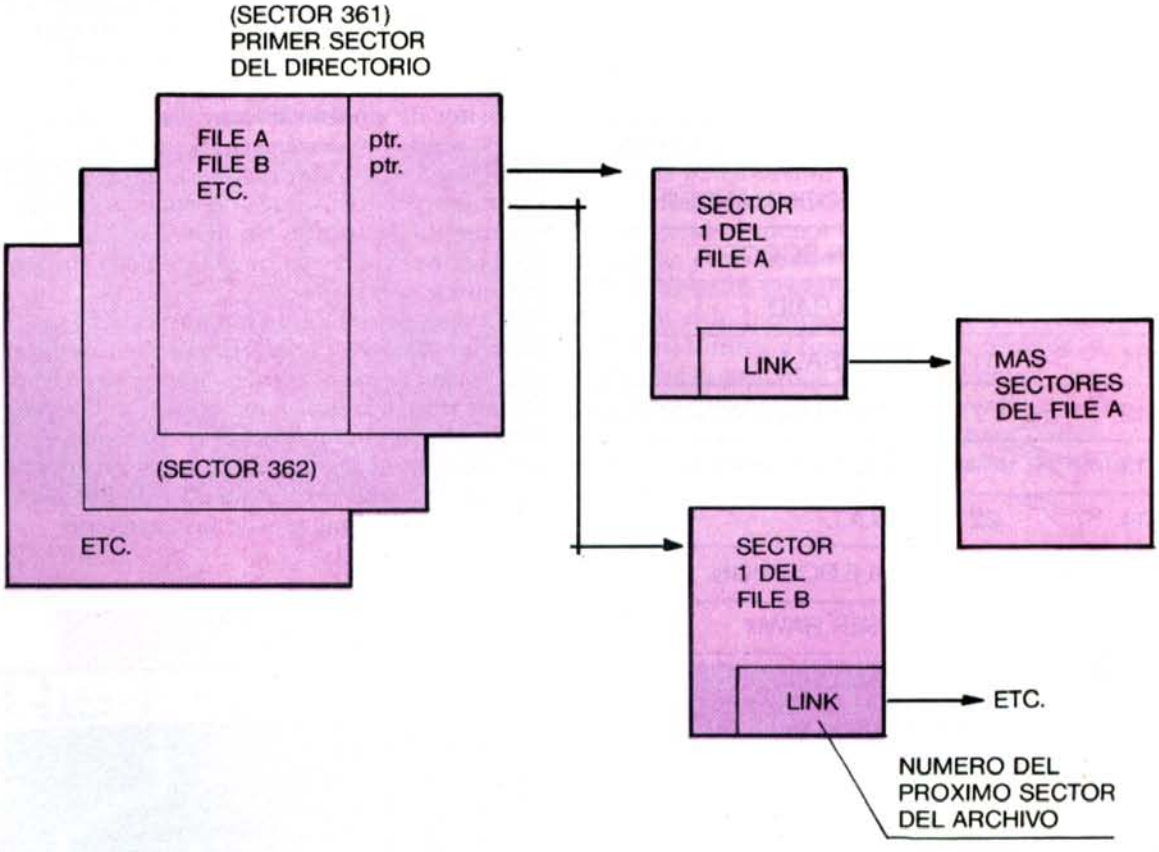
-SECTORES DE INFORMACION

De los 128 bytes por sector, los primeros 125 están reservados para el uso del usuario. Los bytes 125, 126 y 127 son utilizados por el administrador del disco (recuerda que los bytes se cuentan de 0 a 127=128 bytes). La cantidad de bytes del presente sector se encuentra en el byte 127. Este valor puede variar de 0 (sector vacío) hasta 125 (sector lleno).

Los 6 bits de la izquierda del byte 125, contienen el número de archivo al cual pertenece ese sector y se corresponde con la posición del archivo dentro del directorio. Este número puede variar de 0 a 63. Estos bits sirven para garantizar que los sectores de un archivo no se mezclen con los sectores de algún otro archivo.

Los 2 bits que sobran del byte 125, junto con los 8 del byte 126, son utilizados para apuntar al próximo sector del mismo archivo. Si este valor es 0, significa que el presente es el último sector del archivo.

Recuerda una vez más que si estás utilizando las rutinas del CIOV dadas en el número anterior, al no utilizar el Administrador de Diskette puedes disponer de los 128 bytes de cada sector.



CompuCenter

- ATARI
- COMMODORE
- APPLE

ATENCION TODOS
LOS DIAS DEL AÑO

PARQUE ARAUCO
LOCAL 247 - A
FONO: 2420596

- Equipos
- Suministros
- Software
- Materiales didácticos
- Programas IBM-MACINTOSH

RANKING DEL MES

POS. DEL MES	POS. DEL ANTERIOR	TITULO
1	3	HENRY'S HOUSE
2	2	MONTEZUMA'S REVENGE
3	-	THE EXTIRPATOR
4	1	GHOSTCHASER
5	-	TARZAN
6	5	NINJA
7	4	INTERNATIONAL KARATE
8	9	TANK COMMANDER
9	-	SUPER SOCCER
10	12	RIVER RAID
11	11	FOOTBALL
12	7	SCREEMING WINGS 1942
13	-	TWILIGHT WORLD
14	22	S.W.A.T.
15	8	POLE POSITION
16	6	LASER HAWK
17	-	TABLE SOCCER
18	15	BLUE MAX
19	-	LITTLE DEVIL
20	13	BRUCE LEE
21	10	BMX SIMULATOR
22	21	RESCUE ON FRACTALUS
23	23	GHOSTBUSTERS
24	25	FLAK
25	17	GREAT AMERICAN RACE

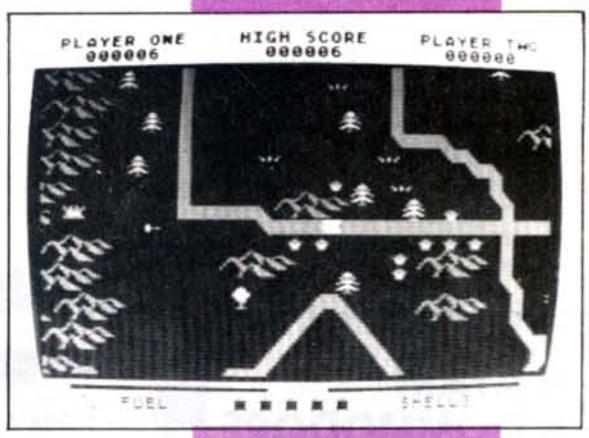
ASCENSO CONSTANTE DESCENSO

Este es el ranking correspondiente al mes de Diciembre, obtenido en base a las estadísticas de ventas de cassettes Turbo Software en todo Chile. Recuerda que tus preferencias también serán tenidas en cuenta, para lo cual debes escribir a Holanda 2456, Providencia, con los juegos de tu elección.

Tank commander

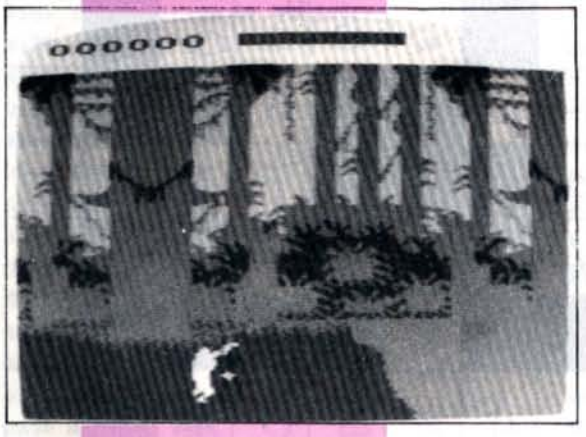
Es el año 1942 y la Segunda Guerra Mundial se encuentra en un punto crítico. Tú eres el Comandante de una escuadrilla de tanques y debes internarte tras las líneas enemigas para destruir sus armas y fuentes de abastecimiento. En cada viaje tendrás una misión distinta, que puede variar desde destruir tanques enemigos, eliminar los depósitos de combustible, acabar con las bases, etc. En el camino habrá cañones anti-tanques, zonas minadas, tanques enemigos y aviones que con vuelos razantes tratarán de destruirte. No debes atacar las poblaciones civiles ya que ello te ocasionará puntos en contra.

Además del manejo del tanque, debes manejar el cañón. Es decir, para enfrentar al aguerrido enemigo, puedes optar entre girar el tanque o bien únicamente el cañón. Cuando debas cruzar cualquiera de los ríos que abundan en la región, debes tener en cuenta que, debido a su profundidad, sólo puedes hacerlo por los puentes de las carreteras.



SONIDO	6.5
GRAFICACION	6.2
ADICCION	6.4
PRESENTACION	6.2
PROMEDIO	6.325

SONIDO	6
GRAFICACION	6.5
ADICCION	6.5
PRESENTACION	6.4
PROMEDIO	6.35



Tarzán

Un grupo de peligrosos tráficantes de animales ha llegado a la selva, capturando monos para venderlos a los zoológicos del mundo. Los cazadores furtivos no se detendrán por nada, así que Tarzán debe recurrir a todos sus conocimientos de la selva para poder liberarlos.

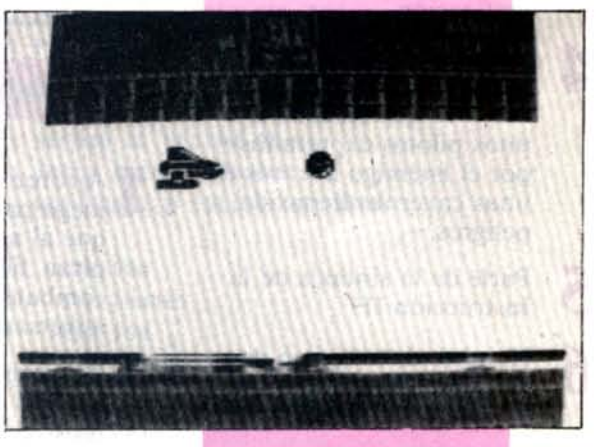
Los peligros que allí abundan son interminables, víboras venenosas prestas a morderlo al menor descuido, gigantescos gorilas protegiendo sus territorios, hambrientos cocodrilos infestando las aguas de los ríos, trampas en el suelo colocadas por los maleantes y finalmente los propios bandidos armados hasta los dientes.

Para enfrentarlos Tarzán cuenta con sus poderosos puños, su agilidad para correr, nadar, trepar a los árboles y saltar de liana en liana.

The Extirpator

Te encuentras en manos de una flotilla de naves espaciales, en un misterioso planeta de una lejana galaxia, habitado por extraños seres sumamente hostiles que intentarán destruirte. Las misteriosas criaturas tienen dos maneras de enfrentarte: tratando de chocarte o disparando sus armas de diversa potencia. Los seres tienen distintas resistencias a tus armas, necesitando algunos de ellos ser impactados varias veces antes de ser destruidos.

Cuentas con un completo panel de instrumentos que te indicarán las averías de tu nave, la cantidad de municiones que te quedan, la cantidad de naves que restan de la fuerza de combate y finalmente en qué nivel te encuentras. Cuanto más te vas introduciendo en sus defensas, tanto más complicado se tornará tu avance.



SONIDO	6.7
GRAFICACION	6.7
ADICCION	6.4
PRESENTACION	6.4
PROMEDIO	6.55

TURBO PUZZLE:

Horizontales:

1 Marca de un computador de 8 bit que actualmente se vende en el mercado.

2 Instrucción Basic que junto al READ se utiliza para almacenar y leer datos directamente en un programa.

3 Lenguaje de Programación orientado a la Educación que utiliza una tortuga para su graficación.

4 Nombre de un juego para nuestro Atari, en el cual tenemos que rescatar a unos pilotos capturados por el enemigo a través de unas cavernas llenas de peligros.

5 Parte de la sintaxis de la Instrucción IF.

6 Nombre de un juego de Ajedrez para nuestro Atari.

7 Los diskettes están divididos en pistas y cada pista en...

8 Instrucción Basic utilizada para acceder al testeo del computador (Self Test).

Verticales:

1 Ajedrez del Futuro. Es un juego para Atari, en el que al atacar a una pieza, tienes que entrar en combate con tu adversario.

9 Nombre de un juego para el Atari en el cual debes recorrer unas sinuosas cavernas.

10 Versión moderna del juego Pong, en el cual deberás tratar de destruir unas paredes multicolores haciendo rebotar una pelotita con tu paleta.

11 Instrucción Basic utilizada para alterar la secuencia normal del programa y dirigirse incondicionalmente a una nueva línea de instrucción.

12 Nombre del juego de la serie televisiva protagonizada por Diego de la Vega y su caballo Tornado.

13 Nombre de los programas que hacen que el computador reconozca y utilice a las disketteras.

6 Nombre del sistema operativo para cassette.

14 Iniciales del Sistema Operativo Atari.

ASSEMBLER

```
0100 ;SAVEHD:EDITOR.MAC
0110 ;ASM,,HD:EDITOR.ASM
0120 ;
0130      .OPT NO LIST
0140      .OPT OBJ
0150 ;
0160 ; DEFINICION DE VARIABLES
0170 ;
0180 FINTABLA = $8000
0190 LINEA = $9000
0200 INBUFF = $F3
0210 CIX =    $F2
0220 FR0 =    $D4
0230 POINTERAUX = $A2 ;A3
0240 LBUFF = $0580
0250 CANTIDAD = $A4
0260 FINTAB = $A5
0270 LINUM = $A7
0280 AUX =    $A9
0290 AUX1 =   $AB
0300 CANT =   $AA
0310 GUARDOX = $AD
0320 GUARDOY = $AE
0330 GUARDOA = $B1
0340 POINTERAUX1 = $AF
0350 POINT = $B2
0360 CORREDOR = $B4
0370 HUECO = $B6
0380 PUNCON = $B7
0390 FINAUX = $B9
0400 MANDO = $BB
0410 POINTER2 = $BD
0420 PUNCAR = $C0
0430 POINTCARTEL = $C2
0440 PP  =    $C4
0450 PUNTABLA = $C6
0460 PORTA = $D300
0470 PACTL = $D302
0480 DDA =    $D300
0490 CRA =    $D302
0500 AUDF2 = $D202
0510 AUDF1 = $D200
0520 STIMER = $D209
0530 IRQEN = $D20E
0540 SHADOWIRQEN = $10
0550 SKCTL = $D20F
0560 SSKCTL = $0232
0570 AUDCTL = $D208
0580 EOT =    $C8
0590 BYTE =   $C9
0600 NMIEN = $D40E
0610 POINTER = $CA
0620 SET =    $CC
0630 UBISET = $CE
0640 PG0 =    $D0
0650 ;
0660 ; DIRECCIONES DE EJECUCION
0670 ;
0680      *=  $02E0
0690      .WORD COMIENZO
0700      *=  $2800
0710 FONTDEF
0720      *=  $3000
0730 ;
0740 ; DIRECCION DE INICIO DEL
0750 ; PROGRAMA.
0760 ;
0770 COMIENZO
0780      LDA #0
0790      STA 710
0800 ;
0810 ;CAMBIO EL SET DE CARACTERES
0820 ;PARA QUE EXISTAN ACENTOS
0830 ;
0840      JSR FONT
0850 ;
0860 ; DEFINO AL VECTOR DEL DOS
0870 ; PARA QUE AL INGRESAR EL
0880 ; COMANDO DOS, SALTE AL DOS
0890 ;
0900 ; DEFINO AL VECTOR DE LA
0910 ; TECLA BREAK PARA QUE AL
0920 ; PRESIONARLA EJECUTE UN STOP
0930 ;
0940      LDA $0A
0950      STA RUNDOS
0960      LDA $0B
0970      STA RUNDOS+1
0980      LDA $09
0990      STA DOS9
1000      LDA # <BREAK
1010      STA $0236
1020      LDA # >BREAK
1030      STA $0237
1040      LDA #0
1050      STA $0244
1060      LDA #2
1070      STA $09
1080      LDA # <INTDOS
```

```
1090      STA $02
1100      LDA # >INTDOS
1110      STA $03
1120      JMP INICIO
1130 ;
1140 ; RUTINA DE ATENCION A LA TECLA
1150 ; BREAK. AL PRESIONARLA EL PRO-
1160 ; GRAMA SALTA A ESTA INSTRUCCION
1170 ;
1180 BREAK
1190      LDA #0
1200      STA FLAGNUM
1210      LDA #$9B
1220      JSR $F2B0
1230      JSR $F2B0
1240      LDA #'G
1250      JSR $F2B0
1260      LDX #$FF
1270      TXS
1280      CLI
1290      JMP IMPLIST
1300      RTI
1310 ;
1320 INICIO
1330      LDA #0
1340      STA FLAGNUM
1350 ;
1360 ; BORRO LA PANTALLA E IMPRIMO
1370 ; LA PRESENTACION
1380 ;
1390      LDA #'5
1400      JSR $F2B0
1410      LDA # <ROTW
1420      STA MANDO
1430      LDA # >ROTW
1440      STA MANDO+1
1450      JSR IMPRIMO
1460      LDA # <TABLA
1470      STA FINTAB
1480      LDA # >TABLA
1490      STA FINTAB+1
1500      LDY #0
1510      STY GUARDOY
1520 NUEVALINEA
1530      LDA # >FONTDEF
1540      STA 756
1550      LDA #0
1560      STA $02B6
1570      LDA FLAGNUM
1580      BNE NUMERACION
1590      JMP SINUMERACION
1600 ;
1610 ; NUMERACION AUTOMATICA
1620 ; DE LINEAS EN PANTALLA
1630 ;
1640 NUMERACION
1650      CLC
1660      LDA DESDENUM
1670      STA VUELTA
1680      ADC INCRENUM
1690      STA DESDENUM
1700      STA FR0
1710      LDA DESDENUM+1
1720      STA VUELTA+1
1730      ADC INCRENUM+1
1740      STA DESDENUM+1
1750      STA FR0+1
1760      LDA FLAGBOL
1770      BEQ NODIOVUELTA
1780      LDA FR0+1
1790      CMP VUELTA+1
1800      BEQ VE00
1810      BCC DIOVUELTA
1820      JMP NODIOVUELTA
1830 VE00
1840      LDA FR0
1850      CMP VUELTA
1860      BCC DIOVUELTA
1870 NODIOVUELTA
1880      LDA #1
1890      STA FLAGBOL
1900      LDA #$00
1910      STA FLAGESTA
1920      JSR EXISTELIN
1930      LDA FLAGESTA
1940      BEQ NUMLINOK
1950 ;
1960 ; IMPRIMO EN LA PANTALLA
1970 ; EL ERROR DE LINEA EXISTENTE
1980 ;
1990      LDA # <LINEAEX
2000      STA MANDO
2010      LDA # >LINEAEX
2020      STA MANDO+1
2030      JSR IMPRIMO
2040      JMP BREAK
2050 ;
2060 ; IMPRIMO EN LA PANTALLA
2070 ; NUMERO DE LINEA EXCEDIDO
2080 ;
2090 DIOVUELTA
2100      LDA # <ROTVUELTA
2110      STA MANDO
2120      LDA # >ROTVUELTA
```

```
2130      STA MANDO+1
2140      JSR IMPRIMO
2150      JMP BREAK
2160 ;
2170 ; GENERO EL NUMERO DE LINEA
2180 ; Y LO IMPRIMO EN LA PANTALLA
2190 ;
2200 NUMLINOK
2210      JSR $D9AA
2220      JSR $D8E6
2230      LDY #0
2240      STY GUARDOY
2250 LOPNUM
2260      LDA LBUFF,Y
2270      BMI ULTIMO
2280      STA LINEA,Y
2290      JSR $F2B0
2300      LDY GUARDOY
2310      INY
2320      STY GUARDOY
2330      JMP LOPNUM
2340 ULTIMO
2350      AND #$7F
2360      STA LINEA,Y
2370      STY GUARDOY
2380      JSR $F2B0
2390      LDY GUARDOY
2400      INY
2410      STY GUARDOY
2420      LDA #32
2430      STA LINEA,Y
2440      JSR $F2B0
2450      INC GUARDOY
2460 ;
2470 ; TOMO UNA NUEVA LINEA PARA
2480 ; PROCESARLA EN CASO DE NO
2490 ; SER UN RETURN
2500 ;
2510      JSR TOMOLINEA1
2520      LDY GUARDOY
2530      LDA LINEA,Y
2540      CMP #$9B
2550      BNE NORETURN
2560      LDA #0
2570      STA FLAGNUM
2580      JMP IMPLIST
2590 NORETURN
2600 ;
2610 ; ELIMINO LOS ESPACIOS EN
2620 ; BLANCO A LA IZQUIERDA DE
2630 ; LA INSTRUCCION
2640 ;
2650      JSR BLANCIZ
2660      JMP SINUMERO
2670 SINUMERACION
2680 ;
2690 ; TOMO UNA NUEVA LINEA PARA
2700 ; PROCESARLA EN CASO DE NO
2710 ; SER UN RETURN
2720 ;
2730      JSR TOMOLINEA
2740      LDA LINEA
2750      CMP #$9B
2760      BNE BLANCK
2770      LDA #0
2780      STA FLAGNUM
2790      JMP IMPLIST
2800 BLANCK
2810 ;
2820 ; ELIMINO ESPACIOS EN BLANCO
2830 ; A LA IZQUIERDA DE LA INSTRUC-
2840 ; CION
2850 ;
2860      JSR BLANCIZ
2870 ;
2880 ; DETERMINO SI LA LINEA ES
2890 ; UN COMANDO O BIEN UNA
2900 ; INSTRUCCION SEGUN TENGA O NO
2910 ; NUMERO DE LINEA
2920 ;
2930      LDA LINEA
2940      CMP #'0
2950      BCC COMANDO
2960      CMP #':
2970      BCS COMANDO
2980 SINUMERO
2990      JSR NUMEROLINEA
3000      BCC NUMEROK
3010      LDA # <ROTVUELTA
3020      STA MANDO
3030      LDA # >ROTVUELTA
3040      STA MANDO+1
3050      JSR IMPRIMO
3060      JMP NUEVALINEA
3070 NUMEROK
3080 ;
3090 ; COMPRIMO LA INSTRUCCION
3100 ; PARA QUE OCUPE MENOR ESPACIO
3110 ; Y LA UBICO EN LA TABLA DE
3120 ; INSTRUCCION
3130 ;
3140      JSR COMPRIMO
3150      JSR PONGOTABLA
3160 ;
```

```
3170 ; TOMO UNA NUEVA INSTRUCCION
3180 ; DEL TECLADO
3190 ;
3200      JMP NUEVALINEA
3210 ;
3220 ; IMPRIMO EL MENSAJE LIST
3230 ;
3240 IMPLIST
3250      LDA #0
3260      STA 710
3270      LDA # <ROTLIST
3280      STA MANDO
3290      LDA # >ROTLIST
3300      STA MANDO+1
3310      JSR IMPRIMO
3320      JMP NUEVALINEA
3330 ;
3340 ; SI LA INSTRUCCION FUE UN
3350 ; COMANDO LO ANALIZO PARA
3360 ; VER SI ES VALIDO Y SI LO
3370 ; ES LO EJECUTO
3380 ;
3390 COMANDO
3400      JSR ANALIZOCMD
3410      LDA #$9B
3420      JSR $F2B0
3430      JMP IMPLIST
3440 AUTONUM
3450      LDA #$FF
3460      STA FLAGNUM
3470      JSR TOMOPARAM
3480      SEC
3490      LDA DESDE
3500      SBC INCREMENTO
3510      STA DESDENUM
3520      LDA DESDE+1
3530      SBC INCREMENTO+1
3540      STA DESDENUM+1
3550      LDA INCREMENTO
3560      STA INCRENUM
3570      LDA INCREMENTO+1
3580      STA INCRENUM+1
3590      LDA #0
3600      STA FLAGBOL
3610      RTS
3620 DOS
3630      LDA RUNDOS
3640      STA $0A
3650      LDA RUNDOS+1
3660      STA $0B
3670      LDA DOS9
3680      STA $09
3690      JMP ($FFFC)
3700 DESDENUM
3710      .WORD 0
3720 INCRENUM
3730      .WORD 0
3740 FLAGBOL
3750      .BYTE 0
3760 FLAGNUM
3770      .BYTE 0
3780 ROTLIST
3790      .BYTE "LISTAR",$9B,"*"
3800 ROTOVER
3810      .BYTE $9B,"MEMORIA EXCEDIDA"
,$9B,"*"
3820 ROTINV
3830      .BYTE $9B,"COMANDO ILEGAL",$
9B,"*"
3840 VUELTA .WORD 0
3850 ROTVUELTA
3860      .BYTE $9B,"NUMERO EXCEDIDO"
3870      .BYTE $9B,"*"
3880 LINEAEX
3890      .BYTE $9B,"LINEA EXISTENTE"
3900      .BYTE $9B,"*"
3910 ;
3920 ; DEFINICION DE LOS COMANDOS
3930 ; VALIDOS CON SU LONGITUD
3940 ; Y LA DIRECCION DE MEMORIA
3950 ; EN DONDE SE UBICA LA RUTINA
3960 ; QUE LO EJECUTA
3970 ;
3980 CMD
3990      .BYTE 6,"CARGAR"
4000      .WORD LOAD
4010      .BYTE 1,"C"
4020      .WORD LOAD
4030      .BYTE 6,"GRABAR"
4040      .WORD FILESAV
4050      .BYTE 1,"G"
4060      .WORD FILESAV
4070      .BYTE 6,"LISTAR"
4080      .WORD LIST
4090      .BYTE 1,"L"
4100      .WORD LIST
4110      .BYTE 5,"NUEVO"
4120      .WORD NEW
4130      .BYTE 9,"RENUMERAR"
4140      .WORD REN
4150      .BYTE 1,"R"
4160      .WORD REN
4170      .BYTE 3,"REM"
4180      .WORD REN
```

```
4190      .BYTE 7,"NUMERAR"
4200      .WORD AUTONUM
4210      .BYTE 3,"NUM"
4220      .WORD AUTONUM
4230      .BYTE 6,"BORRAR"
4240      .WORD DELETEO
4250      .BYTE 1,"B"
4260      .WORD DELETEO
4270      .BYTE 6,"ENVIAR"
4280      .WORD ENVIAR
4290      .BYTE 1,"E"
4300      .WORD ENVIAR
4310      .BYTE 3,"DOS"
4320      .WORD DOS
4330      .BYTE 1,"D"
4340      .WORD DOS
4350      .BYTE $FF
4360 CMDCARTEL
4370      .BYTE 2,"P1"
4380      .WORD PAUSA1
4390      .BYTE 2,"P2"
4400      .WORD PAUSA2
4410      .BYTE 2,"P3"
4420      .WORD PAUSA3
4430      .BYTE 2,"P4"
4440      .WORD PAUSA4
4450      .BYTE 2,"P5"
4460      .WORD PAUSA5
4470      .BYTE 2,"P6"
4480      .WORD PAUSA6
4490      .BYTE 2,"P7"
4500      .WORD PAUSA7
4510      .BYTE 2,"P8"
4520      .WORD PAUSA8
4530      .BYTE 2,"P9"
4540      .WORD PAUSA9
4550      .BYTE 3,"P10"
4560      .WORD PAUSA10
4570      .BYTE 3,"P11"
4580      .WORD PAUSA11
4590      .BYTE 3,"P12"
4600      .WORD PAUSA12
4610      .BYTE 3,"P13"
4620      .WORD PAUSA13
4630      .BYTE 3,"P14"
4640      .WORD PAUSA14
4650      .BYTE 3,"P15"
4660      .WORD PAUSA15
4670      .BYTE 6,"PAUSA1"
4680      .WORD PAUSA1
4690      .BYTE 6,"PAUSA2"
4700      .WORD PAUSA2
4710      .BYTE 6,"PAUSA3"
4720      .WORD PAUSA3
4730      .BYTE 6,"PAUSA4"
4740      .WORD PAUSA4
4750      .BYTE 6,"PAUSA5"
4760      .WORD PAUSA5
4770      .BYTE 6,"PAUSA6"
4780      .WORD PAUSA6
4790      .BYTE 6,"PAUSA7"
4800      .WORD PAUSA7
4810      .BYTE 6,"PAUSA8"
4820      .WORD PAUSA8
4830      .BYTE 6,"PAUSA9"
4840      .WORD PAUSA9
4850      .BYTE 7,"PAUSA10"
4860      .WORD PAUSA10
4870      .BYTE 7,"PAUSA11"
4880      .WORD PAUSA11
4890      .BYTE 7,"PAUSA12"
4900      .WORD PAUSA12
4910      .BYTE 7,"PAUSA13"
4920      .WORD PAUSA13
4930      .BYTE 7,"PAUSA14"
4940      .WORD PAUSA14
4950      .BYTE 7,"PAUSA15"
4960      .WORD PAUSA15
4970      .BYTE 2,"D1"
4980      .WORD DESTELL01
4990      .BYTE 2,"D2"
5000      .WORD DESTELL02
5010      .BYTE 2,"D3"
5020      .WORD DESTELL03
5030      .BYTE 2,"D4"
5040      .WORD DESTELL04
5050      .BYTE 2,"D5"
5060      .WORD DESTELL05
5070      .BYTE 2,"D6"
5080      .WORD DESTELL06
5090      .BYTE 2,"D7"
5100      .WORD DESTELL07
5110      .BYTE 2,"D8"
5120      .WORD DESTELL08
5130      .BYTE 2,"D9"
5140      .WORD DESTELL09
5150      .BYTE 3,"D10"
5160      .WORD DESTELL010
5170      .BYTE 3,"D11"
5180      .WORD DESTELL011
5190      .BYTE 3,"D12"
5200      .WORD DESTELL012
5210      .BYTE 3,"D13"
5220      .WORD DESTELL013
```

```
5230    .BYTE 3,"D14"
5240    .WORD DESTELLO14
5250    .BYTE 3,"D15"
5260    .WORD DESTELLO15
5270    .BYTE 9,"DESTELLO1"
5280    .WORD DESTELLO1
5290    .BYTE 9,"DESTELLO2"
5300    .WORD DESTELLO2
5310    .BYTE 9,"DESTELLO3"
5320    .WORD DESTELLO3
5330    .BYTE 9,"DESTELLO4"
5340    .WORD DESTELLO4
5350    .BYTE 9,"DESTELLO5"
5360    .WORD DESTELLO5
5370    .BYTE 9,"DESTELLO6"
5380    .WORD DESTELLO6
5390    .BYTE 9,"DESTELLO7"
5400    .WORD DESTELLO7
5410    .BYTE 9,"DESTELLO8"
5420    .WORD DESTELLO8
5430    .BYTE 9,"DESTELLO9"
5440    .WORD DESTELLO9
5450    .BYTE 10,"DESTELLO10"
5460    .WORD DESTELLO10
5470    .BYTE 10,"DESTELLO11"
5480    .WORD DESTELLO11
5490    .BYTE 10,"DESTELLO12"
5500    .WORD DESTELLO12
5510    .BYTE 10,"DESTELLO13"
5520    .WORD DESTELLO13
5530    .BYTE 10,"DESTELLO14"
5540    .WORD DESTELLO14
5550    .BYTE 10,"DESTELLO15"
5560    .WORD DESTELLO15
5570    .BYTE 12,"FIN DESTELLO"
5580    .WORD FINDESTELLO
5590    .BYTE 2,"FD"
5600    .WORD FINDESTELLO
5610    .BYTE 7,"GRAFICO"
5620    .WORD GRAFICO
5630    .BYTE 2,"GR"
5640    .WORD GRAFICO
5650    .BYTE 11,"FIN GRAFICO"
5660    .WORD FINGRAFICO
5670    .BYTE 2,"FG"
5680    .WORD FINGRAFICO
5690    .BYTE 10,"VELOCIDAD1"
5700    .WORD VELOCIDAD1
5710    .BYTE 10,"VELOCIDAD2"
5720    .WORD VELOCIDAD2
5730    .BYTE 10,"VELOCIDAD3"
5740    .WORD VELOCIDAD3
5750    .BYTE 10,"VELOCIDAD4"
5760    .WORD VELOCIDAD4
5770    .BYTE 10,"VELOCIDAD5"
5780    .WORD VELOCIDAD5
5790    .BYTE 10,"VELOCIDAD6"
5800    .WORD VELOCIDAD6
5810    .BYTE 10,"VELOCIDAD7"
5820    .WORD VELOCIDAD7
5830    .BYTE 10,"VELOCIDAD8"
5840    .WORD VELOCIDAD8
5850    .BYTE 10,"VELOCIDAD9"
5860    .WORD VELOCIDAD9
5870    .BYTE 11,"VELOCIDAD10"
5880    .WORD VELOCIDAD10
5890    .BYTE 11,"VELOCIDAD11"
5900    .WORD VELOCIDAD11
5910    .BYTE 11,"VELOCIDAD12"
5920    .WORD VELOCIDAD12
5930    .BYTE 11,"VELOCIDAD13"
5940    .WORD VELOCIDAD13
5950    .BYTE 11,"VELOCIDAD14"
5960    .WORD VELOCIDAD14
5970    .BYTE 11,"VELOCIDAD15"
5980    .WORD VELOCIDAD15
5990    .BYTE 2,"V1"
6000    .WORD VELOCIDAD1
6010    .BYTE 2,"V2"
6020    .WORD VELOCIDAD2
6030    .BYTE 2,"V3"
6040    .WORD VELOCIDAD3
6050    .BYTE 2,"V4"
6060    .WORD VELOCIDAD4
6070    .BYTE 2,"V5"
6080    .WORD VELOCIDAD5
6090    .BYTE 2,"V6"
6100    .WORD VELOCIDAD6
6110    .BYTE 2,"V7"
6120    .WORD VELOCIDAD7
6130    .BYTE 2,"V8"
6140    .WORD VELOCIDAD8
6150    .BYTE 2,"V9"
6160    .WORD VELOCIDAD9
6170    .BYTE 3,"V10"
6180    .WORD VELOCIDAD10
6190    .BYTE 3,"V11"
6200    .WORD VELOCIDAD11
6210    .BYTE 3,"V12"
6220    .WORD VELOCIDAD12
6230    .BYTE 3,"V13"
6240    .WORD VELOCIDAD13
6250    .BYTE 3,"V14"
6260    .WORD VELOCIDAD14
```

```
6270      .BYTE 3,"V15"
6280      .WORD VELOCIDAD15
6290      .BYTE 4,"SET1"
6300      .WORD SET1
6310      .BYTE 2,"S1"
6320      .WORD SET1
6330      .BYTE 4,"SET2"
6340      .WORD SET2
6350      .BYTE 2,"S2"
6360      .WORD SET2
6370      .BYTE 2,"BO"
6380      .WORD BORRACAR
6390      .BYTE 6,"BORRAR"
6400      .WORD BORRACAR
6410      .BYTE 2,"CI"
6420      .WORD CONIMAGEN
6430      .BYTE 10,"CON IMAGEN"
6440      .WORD CONIMAGEN
6450      .BYTE 2,"SI"
6460      .WORD SINIMAGEN
6470      .BYTE 10,"SIN IMAGEN"
6480      .WORD SINIMAGEN
6490      .BYTE $FF
6500 RUNDOS .WORD 0
6510 DOS9 .BYTE 0
6520 JUMPER .BYTE "  "
6530 ROTW
6540      .BYTE $9B,"LIX/224
     Versi_n 1.0",$9B
6550      .BYTE "(C) 1989          Waldba
um & Dominguez",$9B,$9B,"*"
6560 ;
6570 ; INCLUSION DEL RESTO DE LOS
6580 ; PROGRAMAS QUE CONSTITUYEN
6590 ; EL SISTEMA
6600 ;
6610      .INCLUDE #D:BLANCIZ.MAC
6620      .INCLUDE #D:NUMERO.MAC
6630      .INCLUDE #D:COMPRIMO.MAC
6640      .INCLUDE #D:TABLA.MAC
6650      .INCLUDE #D:ANALCMD.MAC
6660      .INCLUDE #D:LIST.MAC
6670      .INCLUDE #D:GENERAL.MAC
6680      .INCLUDE #D:TOMOLIN.MAC
6690      .INCLUDE #D:EXTAB.MAC
6700      .INCLUDE #D:CONTAB.MAC
6710      .INCLUDE #D:XREN.MAC
6720      .INCLUDE #D:XBORRA.MAC
6730      .INCLUDE #D:SAVE.MAC
6740      .INCLUDE #D:ENVIAR.MAC
6750      .INCLUDE #D:SENDCAR.MAC
6760      .INCLUDE #D:FONT.MAC
6770 TABLACAR
6780 FINTABLACAR = TABLACAR+1948
6790      *=  $7000-6
6800 ROTULOGRABO .BYTE "CN"
6810 CANTIDADLO .BYTE 0
6820 CANTIDADHI .BYTE 0
6830 GUARDOFINTABLO .BYTE 0
6840 GUARDOFINTABHI .BYTE 0
6850 ;
6860 ; A PARTIR DE ESTA DIRECCION
6870 ; SE ALMACENAN LAS INSTRUCCIONES
6880 ; QUE EL USUARIO INGRESE
6890 ;
6900 TABLA
```

```
0100 ;SAVE#D:BLANCIZ.MAC
0110 ;
0120 ; RUTINA QUE ELIMINA ESPACIOS
0130 ; EN BLANCO A IZQUIERDA DE LA
0140 ; LINEA INGRESADA POR EL
0150 ; USUARIO.
0160 ;
0170 BLANCIZ
0180      LDY #0
0190 LOPIZ
0200      LDA LINEA,Y
0210      CMP #32
0220      BNE NOBLANC
0230      INY
0240      JMP LOPIZ
0250 NOBLANC
0260      LDX #0
0270 LOPIZ2
0280      LDA LINEA,Y
0290      CMP #$9B
0300      BEQ FINIZ
0310      STA LINEA,X
0320      INY
0330      INX
0340      JMP LOPIZ2
0350 FINIZ
0360      STA LINEA,X
0370      RTS
```

```
0100 ;SAVE#D:NUMERO.MAC
0110 ;
0120 ; CONVIERTE EL NUMERO DE LINEA
0130 ; EN DOS BYTES UTILIZANDO LAS
0140 ; RUTINAS DEL FLOATING POINT
```

```
0150 ;
0160 NUMEROLINEA
0170     LDA # <LINEA
0180     STA INBUFF
0190     LDA # >LINEA
0200     STA INBUFF+1
0210     LDA #0
0220     STA CIX
0230     JSR $D800    ;ATASC TO FP
0240     CLC
0250     JSR $D9D2    ;FP TO INTERNO
0260     LDA FR0
0270     STA LINUM
0280     LDA FR0+1
0290     STA LINUM+1
0300     RTS
```

```
0100 ;SAVE#D:COMPRIMO.MAC
0110 ;
0120 ; RUTINA QUE ELIMINA DE LA
0130 ; INSTRUCCION EL NUMERO DE
0140 ; LINEA PARA ALMACENARLA EN
0150 ; LA TABLA DE INSTRUCCIONES
0160 ; OCUPANDO MENOS ESPACIO.
0170 ;
0180 COMPRIMO
0190     LDY #0
0200 CICLO1
0210     LDA LINEA,Y
0220     CMP #32
0230     BEQ EMPCOMP
0240     CMP #'0
0250     BCC EMPCOMP1
0260     CMP #':
0270     BCS EMPCOMP1
0280     INY
0290     JMP CICLO1
0300 EMPCOMP
0310     INY
0320 EMPCOMP1
0330     LDX #0
0340 CICLO2
0350     LDA LINEA,Y
0360     CMP #$9B
0370     BEQ FINCOMP
0380     STA LINEA,X
0390     INY
0400     INX
0410     JMP CICLO2
0420 FINCOMP
0430     STX CANTIDAD
0440     RTS
```

La nueva generación de Software para Computadores Atari

ADQUIERALOS EN LOS SIGUIENTES PUNTOS DE VENTAS

• **ANTOFAGASTA:** COOPERCARAB / KW VIDEO / LA ESPAÑOLA • **VIÑA DEL MAR:** FALABELLA VIÑA / INSIS / MPR COMPUTACION • **VALPARAISO:** COMPUTRONIC • **SANTIAGO:** AUDIO BICICLETA INTERNAC / CASA ROYAL / CENTRO ATARI / COMERCIAL ESTADO / COMPUMANQUE / COMPUCENTER / FALABELLA AHUMADA / FALABELLA P. ARAUCO / IMACO / INFOGROUP / PC STORE / PETERSEN / ROLEC / SUPERMERCADOS UNIMARC / TASCO / VIDEO CLUB INTERNACIONAL • **RANCAGUA:** CASA ZUÑIGA • **CURICO:** MULTIHOGAR • **TALCA:** LIBRERIA "EL AHORRO" / MULTICENTRO / VIDEO CLUB CASSAL • **CHILLAN:** CASA EDISON • **CONCEPCION:** COOPERCARAB / DISMAR / DISMAR 2 / EQUS / PHANTER / RAPSODIA / SESCO • **LOS ANGELES:** DISTRIBUIDORA MERINO • **ANGOL:** SCORPIO • **VICTORIA:** CASA SIGMUND • **TEMUCO:** COMERCIAL MANQUEHUE / ESTABLECIMIENTOS GEJMAN / FALABELLA • **PUCON:** ELTIT • **VILLARRICA:** JOYERIA KETTERER • **VALDIVIA:** ELECTROMUSICA • **LA UNION:** IMPORTADORA COSMOS • **OSORNO:** CASA REAL / FOTO EXPRESS • **PUERTO VARAS:** ELECTRO HORN • **PUERTO MONTT:** COMERCIAL MANQUEHUE / DIMARSA • **COYHAIQUE:** FACI HOGAR • **PUNTA ARENAS:** BALFER LTDA.

JUEGOS 2

PARA COMPUTADORES ATARI

SOFTWARE

LA NUEVA GENERACION

TABLE FUTBOL - TWILIGHT WORLD
THE EXTIRPATOR - SUPER SOCCER
LITTLE TEWIL - TARZAN
AMAUROTE - CHIMERA

ES OTRO PRODUCTO M.P.M.

8 JUEGOS

ATARI ES MARCA REGISTRADA DE ATARI CORPORATION

ADQUIERELOS EN LOS SIGUIENTES PUNTOS DE VENTAS

• **ANTOFAGASTA:** COOPERCARAB KW VIDEO LA ESPANOLA • **VINA DEL MAR:** FALABELLA VINA INSIS MPR COMPUTACION • **VALPARAISO:** COMPUTRONIC • **SANTIAGO:** AUDIO BICICLETA INTERNAC CASA ROYAL CENTRO ATARI COMERCIAL ESTADO COMPUMANQUE COMPUCENTER FALABELLA AHUMADA FALABELLA P ARAUCO IMACO INFOGROUP PC STORE PETERSEN ROLEC SUPERMERCADOS UNIMARC TASCO VIDEO CLUB INTERNACIONAL • **RANCAGUA:** CASA ZUNIGA • **CURICO:** MULTIHOGAR • **TALCA:** LIBRERIA "EL AHORRO" MULTICENTRO VIDEO CLUB CASSAL • **CHILLAN:** CASA EDISON • **CONCEPCION:** COOPERCARAB DISMAR DISMAR 2 EQUS PHANTER RAPSODIA SESCO • **LOS ANGELES:** DISTRIBUIDORA MERINO • **ANGOL:** SCORPIO • **VICTORIA:** CASA SIGMUND • **TEMUCO:** COMERCIAL MANQUEHUE ESTABLECIMIENTOS GEJMAN FALABELLA • **PUCON:** ELTIT • **VILLARRICA:** JOYERIA KETTERER • **VALDIVIA:** ELECTROMUSICA • **LA UNION:** IMPORTADORA COSMOS • **OSORNO:** CASA REAL FOTO EXPRESS • **PUERTO VARAS:** ELECTRO HORN • **PUERTO MONTT:** COMERCIAL MANQUEHUE DIMARSA • **COYHAIQUE:** FACI HOGAR • **PUNTA ARENAS:** BALFER LTDA